

Staff Papers Series

STAFF PAPER P82-15

SEPTEMBER 1982

Matrix Generator and Optionals (MGAO):

Users Guide

Howard McDowell



Department of Agricultural and Applied Economics

University of Minnesota
Institute of Agriculture, Forestry and Home Economics
St. Paul, Minnesota 55108

Matrix Generator and Optionals (MGAO):

Users Guide

Howard McDowell

September 1982

Staff papers are published without formal review within the
Department of Agricultural and Applied Economics.

MGAO - Operation Outline

H. McDowell

Table of Contents

	Page
Preface	
I. INTRODUCTION	1
II. THEORETICAL REVIEW	3
III. DATA ENTRY	10
Card Format	11
IV. SAMPLE PROBLEMS	23
Problem One	23
Problem One, Input Explanation	29
Problem One, Output Explanation	35
Problem Two	37
Problem Two Input Explanation	40
Problem Two Dual	46
VI. HEADER CARDS	47
VII. FINAL COMMENTS	50
APPENDIX A. Variable Name	52
APPENDIX B. Flow Chart	54
APPENDIX C. Program Listing	63

Preface

Matrix Generator and Optionals (MGAO) is a computer software package developed by Paul Chang and Terry L. Roe. The program is designed to generate input data for a linear programming problem approximating a non-linear programming problem, submit the generated problem to an optimization package, from which the user receives standard computer output.

This paper results directly from efforts by the author to utilize the program and is the first comprehensive documentation written on the program. It is hoped that this paper will make available a useful computer program to those interested. Criticism and suggestions are welcome.

Terry L. Roe provided a significant contribution in the theoretical section and in the general organization of the paper. Reviews by Jeff Apland, Vernon Eidman, and Boyd Buxton are appreciated.

I. INTRODUCTION

Matrix Generator and Optionals (MGAO) is a fortran computer program developed to generate input matrices for mathematical programming algorithm.[1] Of primary importance is its capacity to generate a linear programming problem approximating a nonlinear programming problem.

Specifically, the program is capable of generating matrices for solving linear approximations of nonlinear programming problems incorporating linear or nonlinear supply and demand functions, linear and nonlinear production functions having multiple inputs, and substitutability in demand.

The program operates in conjunction with Multi Purpose Optimization System, MPOS, a system of mathematical programming algorithms developed for solving optimization problems on CDC 6000/CYBER computers. The system includes various linear programming (LP), integer programming (IP), and quadratic programming (QP) algorithms, and an interface with CDC's APEX, a system designed for solving large scale linear programming problems.[2]

For purposes of exposition each mathematical program may be viewed as being composed of two parts, a nonaugmented and an augmented section. The nonaugmented portion is perhaps best illustrated or characterized by most traditional linear programming problems. Following Intrilligator, this portion of the problem may be stated as "choosing nonnegative values of certain variables so as to maximize or minimize a given linear function subject to a given set of linear inequality constraints....

$$\dots \max_{\underline{X}} F(\underline{X}) = \underline{C}\underline{X}$$

Subject to

$$\underline{A}\underline{X} \leq \underline{b}, \underline{X} \geq 0$$

(where \underline{A} is $m \times n$, \underline{X} , $n \times 1$, \underline{C} , $1 \times n$, \underline{b} , $n \times 1$)

"or, written out in full:

$$\max_{x_1, x_2, \dots, x_n} F(x_1, x_2, \dots, x_n) = c_1 x_1 + c_2 x_2 + \dots + c_n x_n$$

subject to:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2$$

$$\vdots$$

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m$$

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0."$$

Clearly, a problem of this nature requires nothing more than defining the activities (x's) the coefficients (c's and a's) and the right-hand-side (RHS) parameters (b's). Therefore in this respect, MGAO is simply a means of entering the data for a linear programming problem, or the linear portion of a nonlinear programming problem. This specification is referred to as the nonaugmented problem, i.e. it has not been augmented to include a nonlinear function.

The augmented portion of the matrix is that portion generated by the program from input data in linear functional form. The principle involved is that a nonlinear function may be approximated by a number of linear steps each of which is a separate linear programming activity. Hence, this technique is also known as separable programming. As the number of steps increases the loss in accuracy decreases. The nonlinear programming problem is stated by Intrilligator below.

"The nonlinear programming problem is that of choosing nonnegative values of certain variables so as to maximize (minimize) a given quasi-concave (convex) function subject to a set of inequality constraints....

$$\dots \max_{\underline{x}} F(\underline{x}) \text{ subject to } \underline{g}(\underline{x}) \leq \underline{b} \quad \underline{x} \geq \underline{0}$$

or written out in full:

$$\begin{aligned} & \max_{x_1 \dots x_n} F(x_1 \dots x_n) \text{ subject to} \\ & g_1(x_1 \dots x_n) \leq b_1 \\ & g_m(x_1 \dots x_n) \leq b_m \\ & x_1 \geq 0, \dots, x_n \geq 0. \end{aligned} [4]$$

This portion of the problem requires entering the objective function $F(\underline{X})$, and the constraints $g(\underline{X})$, in nonlinear form. MGAO then defines discrete linear programming activities with the appropriate objective and constraint activities according to the instructions provided by the user.

The augmented portion of the matrix is also referred to as the extended portion of the matrix.

In specification of problems with both nonaugmented and augmented matrices, the user is advised to design the matrices such that the nonaugmented portion of the matrix, i.e. that part not containing linear approximations of nonlinear equations, is in the upper left hand portion of the matrix and that all transfer or summary columns from the generated rows of the matrix containing the linear approximations of nonlinear functions be on the left hand side. This will prevent respecification to fit the program input format. This will become apparent with examination of the same problems.

II. THEORETICAL REVIEW

Although the program can be used in solving many different types of problems it was designed to facilitate the solution of sectoral models. The user is referred to Duloy and Norton, and Klein and Roe.[5]

The concept is that given "well-behaved" supply and demand functions, a market equilibrium price and quantity may be found by maximizing the area bounded on the right by the supply and demand curves.

Referring to Figure 1. Equilibrium Solution, the equilibrium price and quantity, p^* , q^* , may be found by discovering the quantity that maximizes (area A and area B).

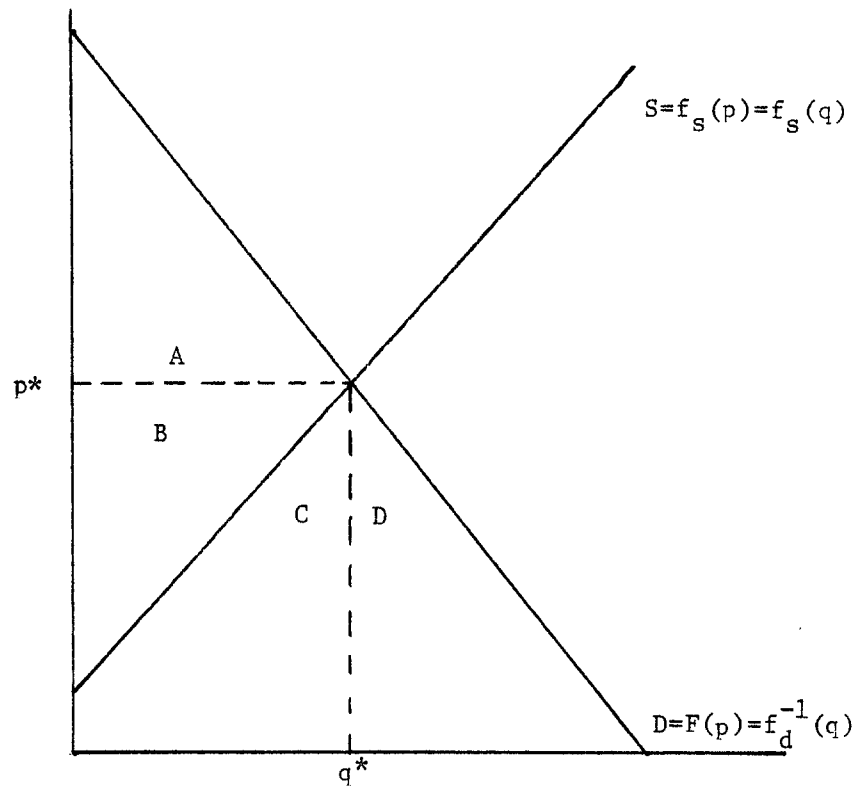


Figure 1. Equilibrium Solution.

Area A is the portion of the integral under the demand curve above p^* , area B is the portion of the rectangle p^*q^* above the integral under the supply function. Under certain conditions these areas are commonly referred to as consumer and producer surplus, respectively. The total area, $A + B = Z$ may be stated as follows:

$$(1) \quad Z = \int_0^{q^*} f_d^{-1}(q) dq - p^*q^* + p^*q^* - \int_0^{q^*} f_s(q) dq$$

where:

$p = f_d^{-1}(q)$ is an inverse demand function,
where price is a function of
quantity demanded; and

$MC = f_s(q)$ marginal cost is a function of the
quantity supplied.

Rearranging the equation for Z,

$$(2) \quad Z = \int_0^{q^*} f_d^{-1}(q) dq - \int_0^{q^*} f_s(q) dq.$$

Provided that Z is a quasi-concave function, is twice continuously differentiable, and in the domain of real numbers, q^* may be found by maximizing Z with respect to q.

Applying the Kuhn-Tucker theorem, the necessary conditions for a maximum to exist are stated as follows:

$$(3) \quad \frac{\partial Z}{\partial q} = f_d^{-1}(q) - f_s(q) \leq 0 \text{ and } \frac{\partial Z}{\partial q} q = 0$$

Rearranging, $f_d^{-1}(q) = f_s(q)$

Substituting p for $f_d^{-1}(q)$ and MC for $f_s(q)$ results in the competitive solution of price and marginal cost being equal.

One may easily complicate this problem by moving to an interregional trade problem. Similarly, the total cost function, $\int_0^q f_s(q) dq$ could be replaced with input supply functions and a production function.

Following Klein and Roe, the following simple nonlinear programming problem is specified, and then converted to a linear programming problem. Both equation and tableau specifications are provided for the linear problem. For a simple case, derivation of the economic information embodied in the dual variables of the LP problem is provided.[6]

Let the demand function for the j^{th} commodity, $j = 1, \dots, J$, be specified in inverse form as:

$$(4) \quad p_j = a_j - b_j q_j$$

where q_j is the quantity demanded, a_j is the intercept, and b_j is the change in the quantity of q_j demanded given a change in its own price, p_j .

Let the supply side be specified by the following total cost and conversion equations.

$$(5) \quad \text{Let } q_j = m_j x_j, \quad j = 1, \dots, J$$

where $m > 0$ is the conversion factor for x into q .

Let c_j be the unit cost of x_j , $j = 1, \dots, J$.

The nonlinear programming specification of this problem is

$$(6) \quad \max_{q, x} Z = \sum_j \int_0^{q_j} (a_j - b_j q_j) dq_j - \sum_j c_j x_j + \sum_j \lambda_j (m_j x_j - q_j)$$

or in matrix form,

$$(7) \quad \max_{Q, X} Z = Q'(A - .5BQ) - C'X + \lambda((MX)' - Q')$$

where:

Q is $J \times 1$ of elements q_j

B is $J \times J$ of elements b_j

C is $J \times 1$ of elements c_j

X is $J \times 1$ of elements x_j

λ is $J \times 1$ of elements λ_j

M is $J \times J$ of elements m_{ij} , all $m_{ij} = 0$ for $i \neq j$.

The procedure for linearizing the problem is to find the definite integral of each of the j demand equations,

$$p_j = a_j - b_j q_j, \text{ and evaluate the integrals for } q_j \text{ varying over } i, \text{ or } q_{ji}, i=1, \dots, I, \text{ over } j = 1, \dots, J, \text{ or}$$

$$w_{ji} = a_j q_{ji} - 0.5 b_j q_{ji}^2$$

For each commodity, q_j , the area under its demand curve is found for $i=1$ to I steps. Each of these steps, w_{ji} , are to be activities in the linear programming format, and enter the solution at levels a_{ji} . Certain restrictions (to be explained) are placed on the a_{ji} in order to insure feasibility.

The linear programming problem may be stated as follows:

$$(8.1) \max_{a, x} Z^0 = \sum_{ji} a_{ji} w_{ji} - \sum_j c_j x_j$$

Subject to the J commodity balance constraints

$$(8.2) m_j x_j - \sum_i a_{ji} q_{ji} \geq 0 \quad j=1, \dots, J,$$

and J convexity constraints,

$$(8.3) \sum_i a_{ji} \leq 1 \quad j=1, \dots, J$$

$$\text{or } \max_{a, x} Z^0 = \sum_{ji} a_{ji} w_{ji} - \sum_j c_j x_j + \sum_j \lambda_j (m_j x_j - \sum_i a_{ji} q_{ji}) + \sum_j \lambda_j^* (1 - \sum_i a_{ji})$$

This problem is shown in tableau form below in Table 1..

The convexity constraints are crucial to the problem. Duloy and Norton have shown that if the nonlinear problem is concave, a nontrivial solution will exist where the following will hold for each of the j activities. Either,

- (a) $a_{ji} = 1$, all other $a_{js} = 0$ for a particular j ,
- (b) $a_{ji} < 1$, all other $a_{js} = 0$ for a particular j , or
- (c) $a_{ji} + a_{j(i+1)} = 1$ and all other $a_{js} = 0$, $s \neq i, i+1$. [7]

Table 1. Specification of Commodity Market Demand in Linear Programming Format.

Constraint Constants			Supply Activities (x)	Demand Activities (λ)	Dual
Commodity Balance	0	\leq	m	$-q_1 -q_2 \dots -q_I$	Market Price (π)
Convexity Constraint	1	\geq		1 1 ... 1	Consumer Surplus (π^*)
Objective Function	Z =		- c	$w_1 w_2 \dots w_I$	Consumer Plus Producer Surplus

Source: Klein, Harold E. and Terry L. Roe, "Agriculture Sector Analysis Model Design: The Influence of Administrative Infrastructure Characteristics," Table A.1, p. 299.

The implication is that depending on the difference between segments q_{ji} and $q_{j(i+1)}$ the solution to the linear problem, Z° , can be shown to be an arbitrarily close approximation of the solution to the nonlinear problem Z .

Given this arbitrary closeness of the linear to the nonlinear problem, it can be shown that the duals of commodity balance rows are equal to the prices, and that the duals of convexity constraints are equal to consumer surpluses. Case (a) is used for simplicity, otherwise the problem is complicated by combination of a_{ji} , or fractional values of a_{ji} .

For a positive a_{ji} , the Kuhn-Tucker conditions require that,

$$\frac{\partial Z^\circ}{\partial a_{ji}} = w_{ji} - \lambda_j q_{ji} - \lambda_j^* = 0.$$

For a basis variable, it follows from the nonlinear problem that

$$\begin{aligned} \frac{\partial Z}{\partial q_{ji}} &= \frac{\partial (w_{ji})}{\partial q_{ji}} - \lambda_j = 0 \\ &= p_{ji} - \lambda_j = 0. \end{aligned}$$

Therefore λ_j , the shadow price or dual for the commodity balance row is equal to the equilibrium commodity price.

Since a_{ji} is assumed to be one, and Z° is an approximation of Z , p_{ji} may be substituted for λ_j and

$$\frac{\partial Z^\circ}{\partial a_{ji}} = w_{ji} - p_{ji} q_{ji} - \lambda_j^* = 0.$$

That is, λ_j^* , the shadow price on the convexity constraint is shown to be the consumer surplus for q_{ji} at p_{ji} .

These results can be extended to the production side in the case of total cost expressed as an integral of marginal cost instead of average cost times quantity. In the case of production functions, it is asserted that the shadow prices on the convexity constraints are producer surpluses accrued to the holder of the processes.

It should be pointed out that fixed factors having a positive opportunity cost are also included in calculations of other relevant shadow prices. The same is true for any other form of price or quantity restriction. In order to determine exactly what is involved in the determination of a dual value, Kuhn-Tucker conditions should be stated for each problem, from which expressions for all dual values may be derived.

In summary:

1. The shadow prices on commodity balance constraints for demand functions are equilibrium market prices for the commodities.
2. The shadow prices on convexity constraints for demand functions are consumer surpluses associated with the commodities.
3. The shadow prices on factor balance constraints for supply functions are equilibrium market prices for the factors.
4. The shadow prices on convexity constraints for supply functions are producer surpluses associated with the factors.
5. The shadow prices on convexity constraints for production functions are producer surpluses associated with production of the commodities.

III. DATA ENTRY

In proceeding to the section explaining the data entry it should be useful for the user to have a broad view of how the program operates.

The first block of information includes the dimensions of the nonaugmented portion of the matrix, the algorithm and/or system desired (one of several MPOS algorithms or APEX). The objective function, constraints, and if an integer program, the integer variables are read in.

The second possible block of information is in conjunction with an option to read in a second data set to be inserted some place within the data set previously read in for the initial

models. This option could be useful in the case of expanding the number of columns or rows somewhere in the middle of the nonaugmented portion of the matrix, without having to repunch a new data deck.

The third possible block of data includes the information necessary to generate linear activities approximating a nonlinear function. This block is further divided into two groups of functions and associated procedures.

The simpler of the two entails taking linear steps of a single variable function, and calculating the coefficients for the objective function and the row constraints. Examples of this type of function include supply and demand curves where quantity is a function of price. The program calculates the area under the curve at each quantity increment. These values are then placed into the objective and appropriate constraint specification by the algorithm.

The more complex of the two nonlinear functions involves the generation of an input substitution surface. An isoquant defining the relationship of an output, Q , two inputs X_1 , and X_2 , in Cobb-Douglas functional form is provided for. It is also conceivable that if Q were viewed as a composite consumption good, the surface could represent how X_1 and X_2 substitute in the consumption of Q . For example Q could be fruit, X_1 oranges, and X_2 apples, the program will calculate as many activities as necessary to satisfy the steps in Q desired.

Card Format

In moving through the data input cards, the user may wish to refer to the listing of variable names and options, the flow chart, and the program listing found in Appendices A, B, and C, respectively.

Input cards are listed in read statement form, each with its fortran format given. A short explanation is given where program branches occur, or where an explanation may otherwise be helpful.

1. READ (5,500) IDM, M1, COL, ROW

500 FORMAT (I1, I2, 2I5)

IDM =0 for maximum

=1 for minimum

M1, algorithm within MPOS

= 01, - REGULAR -, 2-phase simplex (LP)

= 02, - REVISED -, revised simplex (LP)

= 03, - DUAL -, dual simplex (LP)

= 04, - MINIT -, primal-dual (LP)

= 05, - BBMIP -, branch and bound mixed integer program (IP)

= 06, - DSZLIP -, direct search 0-1 integer program (IP)

= 07, - GOMORY -, Gomory's cutting plane (IP)

= 08, - WOLFE -, Wolfe's quadratic simplex (QP)

= 09, - BEALE -, Beale's algorithm (QP)

= 10, - LEMKE -, Lemke's complementary pivot algorithm (QP)

= 11, - APEX 1 -, MPOS-APEX data file interface (GENERAL)

= 12, - APEX 2 -, MPOS-APEX data file interface (GENERAL)

COL, number of columns in nonaugmented matrix

ROW, number of rows in nonaugmented matrix.

2. READ (5,501) TITLE

501 FORMAT (8A10)

3. If the problem is an integer programming problem, the following cards are punched indicating the number of integer variables and variable names. If the problem is not IP, then the card block is left out.

READ (5,503) N2, (ACT(I1), I1 = 2,N2)

503 FORMAT (I3, 11A7/(3X, 11A7))

N2, the member of integer variables

ACT (I2), the activity names

4. Read in the nonaugmented or traditional LP activities

```
READ (5,505) (ACT(IA), IA = 1, COL)
```

```
505 FORMAT (3X, 11A7)
```

ACT(IA), activity names

COL, number of columns in nonaugmented matrix

5. Read in the nonzero coefficients of the objective function of the nonaugmented matrix. Activities such as transfer columns having no objective value need not be entered.

```
READ (5,506) (ICOL(IB), SIGN(IB), COEF(IB), IB = J1, J1+4)
```

```
506 FORMAT (5(I4, A1, F11.2))
```

ICOL(IB), the integer number of the activity, ACT(IA) for

which an objective value is entered. Numbers

begin with the left hand side of the matrix with

1, and run consecutively up through COL.

SIGN(IB), the sign, + or -, of the objective value

COEF(IB), the real value of the objective function.

Note that up to five such entries may be entered on each card.

FLAG - Once all objective values are read in, or if there are

no nonzero values associated with the nonaugmented matrix,

then ICOL = -999. So at least one card, with entry -999

in the first four columns is necessary if any nonaugmented

activities are entered.

6. Read in the constraints for the nonaugmented matrix.

```
READ (5,506) (ICOL(IF), SIGN(IF), COEF(IF), IF = J2, J2+4)
```

```
506 FORMAT (5(I4, A1, F11.2))
```

Exactly as in the case of the objective function, only the nonzero

coefficients need be entered. In order to signify the completion

of input for each constraint, three possible values may be assigned

to ICOL. These values coincide with the nature of the constraints.

ICOL = -100, ——— \leq RHS constraint

ICOL = -200, ——— = RHS constraint

ICOL = -300, ——— \geq RHS constraint

Just as in the case of the column coefficients, the right hand side parameter is entered with SIGN and COEF along with the appropriate ICOL value. No other indicator is necessary to signify the completion of constraint input.

If this block of cards complete the data input, it is followed by an end-of-file (EOF) card. This card is multiple punched, 7-8-9, in the first column, and completes the input.

7. Read in data for the insertion option.

READ (5,511) ISID

511 FORMAT (I5)

If new activities are to be inserted, ISID is given the value of 99999, and a subroutine called INSERT is called. If the user does not desire to use the insert option, a blank card is necessary.

If the insert option is used, the cards following ISID, and used by the subroutine INSERT are listed below.

1. Location of insertion

READ (5,511) NINS

511 FORMAT (I5)

NINS is the column number of the existing nonaugmented matrix at which the new activities are to be inserted.

2. Number and name of inserted activities

READ (5,503) NAA, (AACT(IA), IA = 1, NAA)

503 FORMAT (I3, 11A7/(3X, 11A7))

NAA, the number of new activities to be inserted.

AACT, names of the new activities.

3. Read in objective of inserted activities.

```
READ (5,506) (AICOL(IB), ASIGN(IB), ACOEF(IB), IB = 1,NAA)
506 FORMAT (5(I4, A1, F11.2))
```

This input is identical in format to the objective data entered above. However unlike the earlier case in which only nonzero coefficients were entered, an objective value for each inserted activity must be entered.

AICOL, the column number of the inserted activity, beginning with 1.

ASIGN, the sign on the coefficient.

ACOEf, the objective coefficient.

4. Read in number of nonzero coefficients to be inserted.

```
READ (5,511) NBB
511 FORMAT (I5)
```

NBB, the number of nonzero constraint coefficients to be inserted.

5. Read in the coefficients

```
READ (5,512) (AEWROW(IX), AEWCOL(IX), AEWSIGN(IX), AEWCoeff(IX),
IX = 1, NBB)
512 FORMAT (4(2I3, A1, F13.2))
```

AEWROW, row number of the coefficient.

AEWCOL, column number of the coefficient.

AEWSIGN, sign of the coefficient.

AEWCoeff, the coefficient.

Note, this option has not been tested and it is unclear whether or not the numbers for AEWROW and AEWCOL are row and column numbers of the new matrix. However, this appears to be the most logical first choice. As above, if this block of data is final, then an EOF card follows the insertion and the input is completed.

7. Read in information for extended functions from which the augmented portion of the matrix is composed.

This section is characterized by having two options. The first is to generate linear activities from a single nonlinear function, such as a supply or demand function, the second is to generate a substitution relationship between 2 variables according to an exponential function, such as a production function with 2 input variables. Data entry is given for both of these cases.

```
READ (5,510) EID, RM, IDPV(JA)
```

```
510 FORMAT (I5, F10.2, I5)
```

IDPV, flag for two variable function,

= 0, single variable function,

≠ 0, three variable function.

EID, for IDPV = 0, denotes the number of nonzero coefficients for activities in the nonaugmented matrix in the same row as the generated activities; for IDPV ≠ 0, EID = 3, denoting the number of rows necessary for the exponential function, one row each for X_1 , X_2 , and Y.

RM, the right-hand-side value for the extended row. This program is designed for the RHS value of an extended row to be either 1.0 or 0.0. For each set of activities generated, a convexity constraint is generated automatically having a RHS value of 1.0. If a RHS value of zero is desired then RM is given a value of zero. Although no example is readily available for which it may be useful, it is possible to enter a negative RHS value but not possible to enter a positive RHS value. In general, RM will be given a value of 0.0.

Case a. Single Variable Function

This type of function will require the use of a single quantity constraint row. In the case of a supply or demand function, the generated activities in the augmented portion of the matrix will require at least one transfer activity in the same row in the nonaugmented portion of the matrix. It is possible, however, to generate augmented activities with no other coefficients in the same row.

In this case, IDPV = 0, EID = K, where K is the number of nonzero coefficients for the row in the nonaugmented portion of the matrix, and RM = 0.0, unless a negative RHS is desired. Note that in the case where all three values equal zero, a blank card is still necessary for the program to proceed.

The following cards are punched in the case of IDPV = 0.

1. If EID \neq 0, read in coefficients, otherwise, skip this card and proceed to 2.

```
READ (5,506) (ICOL(II), SIGN(II), COEF(II), II = 1,EID)
```

```
506 FORMAT (5(I4, A1, F11.2))
```

ICOL, the number of columns in which the coefficient is to be entered.

SIGN, the sign of the coefficient, + or -.

COEF, the coefficient to be entered.

2. Read in mathematical function to be extended.

The program is designed for input of exponential functions of the following form:

$$W = C_1 X^{\alpha_1} + C_2 X^{\alpha_2} + \dots + C_n X^{\alpha_n}.$$

Note that in the case of X being a commodity for which a supply or demand function is defined, the equation entered is the integral of the supply or demand function. In the case of supply, the equation above would represent the total cost function associated with a marginal cost or supply function of the form:

$$\frac{\partial W}{\partial X} = MC = \alpha_1 C_1 X^{\alpha_1 - 1} + \alpha_2 C_2 X^{\alpha_2 - 2} + \dots + \alpha_n C_n X^{\alpha_n - 1}.$$

If the first term were an intercept, α_1 would have the value of 1, so that the value would simply be C_1 .

READ (5,508) IJ, (CC(IK), IEXP(IK), IK = 1,IJ)

508 FORMAT (I5,5(F10.4, F5.0))

IJ, the number of terms in the function.

CC, the coefficients C_i for the function.

IEXP, the exponents α_i for the function.

3. Read in the initial value, the magnitude, and number of steps to be taken in the linearization procedure.

READ (5,509) A(1), DELTA Q, STEP

509 FORMAT (2F10.4, I5)

Q(1), initial value of the function.

DELTAQ, the increment value, $(Q_i - Q_{i-1}) \forall_i = 1, n$.

STEP, the number of steps, n, taken.

From the function and linearization information, the program adds the number of columns consistent with the number of steps, and calculates the area under the function at each step for the objective function. Two constraints are generated, a quantity allocation constraint containing the quantity steps specified, and a convexity constraint.

All quantity steps are generated having negative signs. The direction of the constraint is determined by the program to be, \leq , in the case of a supply function, \geq , in the case of a demand function.

Case b. Multiple Variable Function

As stated above, the most obvious use of this option is to incorporate a production function where two inputs, X_1 and X_2 , are combined in the production of various quantities of some Y , specified by a Cobb-Douglas type function.

The concept used is to define several input ratios, or expansion paths at various levels of Y . From the ratio and Y values, values for X_1 and X_2 are determined. The calculation of the ratios follow:

$$Y = AX_1^{\alpha_1}X_2^{\alpha_2}$$

$$\text{RATIO} = (X_1/X_2), \text{ rearranging}$$

$$X_1 = X_2R, \text{ where } R = \text{RATIO.}$$

Substituting for X_1 , and solving for X_2 .

$$Y = A(X_2R)^{\alpha_1}X_2^{\alpha_2}$$

$$Y = AX_2^{\alpha_1\alpha_1\alpha_2}R^{\alpha_1\alpha_2}$$

$$X_2^{\alpha_1+\alpha_2} = YA^{-1}R^{-\alpha_1}$$

$$X_2 = (YA^{-1}R^{-\alpha_1})^{\frac{1}{\alpha_1+\alpha_2}}$$

For each ratio r_i , $i = 1, \dots, n$, and for varying levels of Y , a unique value of X_2 is calculated which in turn determines the appropriate value of X_1 . This grid linearization is illustrated in Figure 2. The Linearized Specification of $Y = f(X_1, X_2)$.

The number of activities generated in the number of steps in Y times the number of ratios. Three quantity constraint rows, one each for X_1 , X_2 , and Y , and a convexity constraint row are generated by the program. Zeroes are placed in the objective function.

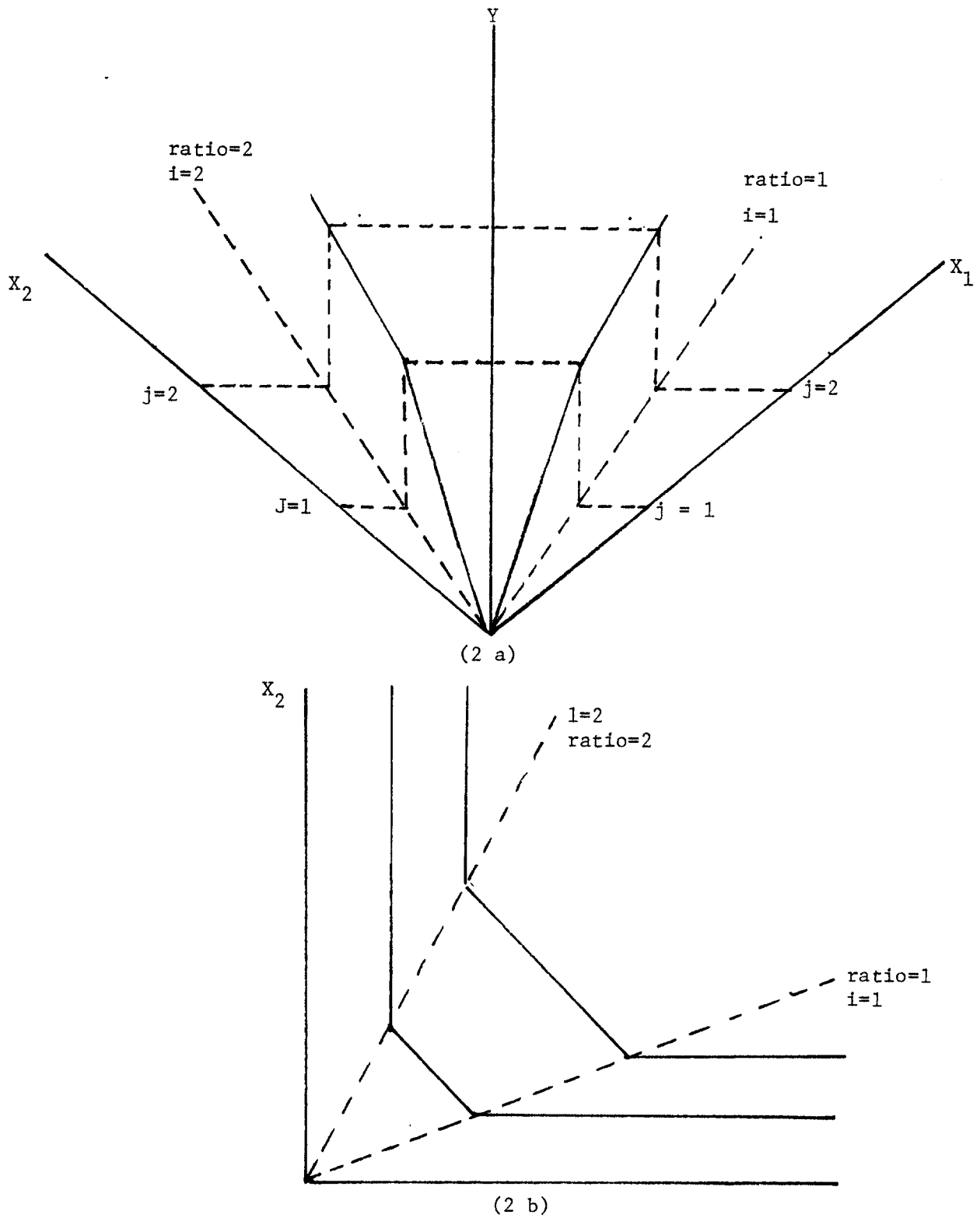


Figure 2. The Linearized Specification of $Y=f(X_1, X_2)$.

Source: Roe, Terry. "Modeling of Nonlinear Functions into A Linear Programming Format". Staff Paper P75-9 [8]

Values for the first card of this group are as follows:

IDPV \neq 0, the value 99999 is given in some sample decks

EID = 3, the number of quantity rows to be generated

RM = 0.0, right hand side values.

For each of the three rows, the following sequence of cards is necessary.

1. Read in the number of coefficients in the row in the nonaugmented portion of the matrix.

READ (5,507) IEID

507 FORMAT (I5).

IEID, the number of coefficients.

2. Read in the column, sign, and value of the coefficients

READ (5,506) (ICOL(I1), SIGN(I1), COEF(I1), I1=1, IEID(I3)+1)

506 FORMAT (5(I4, A1, F11.2))

ICOL, the column number in which the coefficient is to be entered.

SIGN, the sign of the coefficient.

COEF, the value of the coefficient.

Important! Note that the right hand side value and the constraint type must be entered by the user. Therefore the final entry will have one of the following values for ICOL:

ICOL = -100 ——— \leq RHS constraint,

ICOL = -200 ——— = RHS constraint,

ICOL = -300 ——— \geq RHS constraint.

3. Read in information pertaining to the ratios to be used in generating activities.

READ (5,584) NOR,(RATIO(I2), I2=1, NOR)

584 FORMAT (I3, 11F7.2/(3X, 11F7.2))

NOR, the number of ratios.

RATIO, the ratio, (X_1/X_2) .

4. Read in the Cobb-Douglas function parameters

$$Q = AAX_1^{\alpha_1} X_2^{\alpha_2}$$

READ (5,585) AA, ALPH1, ALPH2

585 FORMAT (3F10.2)

AA, the multiplicative coefficient

ALPH1, the exponent on X_1 .

ALPH2, the exponent on X_2 .

5. Read in the initial value, the magnitude, and the number of steps to be taken.

READ (5,509) Q(1), DELTAQ, STEP

509 FORMAT (2F10.4, I5)

Q(1), initial value of the function.

DELTAQ, the increment value.

STEP, the number of steps taken.

This concludes the data input section. It should be pointed out that once the input is complete, end of file (EOF) card is required. This card is punched 7-8-9 in the first column.

IV. SAMPLE PROBLEMS

For illustrative purposes, two sample problems developed by Roe are provided. The first problem is stated in nonlinear form and then restated in linear form. Results concerning the values of shadow prices on commodity balance and convexity constraints are provided.

Provided for both problems are verbal and mathematical specification, tableau representation, data input deck, MPOS specification, and finally MPOS summary of results.

Problem One

The first problem is one of maximizing the sum of producers' and consumers' surplus, with a variety of perfectly inelastic and elastic, and sloping supply and demand functions, and a production function.

Three commodities which are perfectly inelastically supplied may be combined. One of these inputs and an input supplied with an upward sloping function, may be combined to produce another commodity. The produced commodity faces a downward sloping demand.

The nonlinear programming specification of the problem follows:

$$\text{MAX } Z = .5\text{LACT1} + .9\text{LACT2} + .7\text{LACT3} - 1.5X_1$$

$$- \int_0^{X_2} (X_2) dX_2 + \int_0^Y (90 - 1.2Y) dY$$

$$+ \lambda_1 (90 - .4\text{LACT1} - .3\text{LACT2})$$

$$+ \lambda_2 (80 - .3\text{LACT1} - .2\text{LACT3})$$

$$+ \lambda_3 (200 - .4\text{LACT2} - .9\text{LACT3} - X_1)$$

$$+ \lambda_4 (4X_1^3 X_2^5 - Y)$$

where:

X_2 = supply or marginal cost of X_2 , and

$90 - 1.2Y$ = Inverse demand function for Y .

The nonlinear programming problem is now converted to a linear programming problem. Notice that each of the constraints stated in Lagrangian form corresponds exactly to a row constraint in the tableau specification of the problem found below.

$$\begin{aligned}
 \text{Max } Z^0 = & .5LACT1 + .9LACT2 + .7LACT3 - 1.5X_1 - \sum_{m=1}^M a_m \gamma_m + \sum_{n=1}^N a_n w_n \\
 & L, X_1, a \\
 & + \lambda_1 (90 - .4LACT1 - .3LACT2) \\
 & + \lambda_2 (80 - .3LACT1 - .2LACT3) \\
 & + \lambda_3 (200 - .4LACT2 - .9LACT3 - X_1) \\
 & + \lambda_4 (X_2 - \sum_{m=1}^M a_m X_{2m}) \\
 & + \lambda_5 (1 - \sum_{m=1}^M a_m) \\
 & + \lambda_6 (\sum_{p=1}^P a_p X_{1p} - X_1) \\
 & + \lambda_7 (\sum_{p=1}^P a_p X_{2p} - X_2) \\
 & + \lambda_8 (\sum_{p=1}^P a_p Y_p - Y)
 \end{aligned}$$

$$\begin{aligned}
 & + \lambda_9 \left(1 - \sum_{p=1}^P a_p\right) \\
 & + \lambda_{10} \left(Y - \sum_{n=1}^N a_n Y_n\right) \\
 & + \lambda_{11} \left(1 - \sum_{n=1}^N a_n\right)
 \end{aligned}$$

Where:

$\gamma_m = \int_0^{X_2^m} (X_2) dX_2 = [0.5X_2^2]_0^{X_2^m}$, the area under marginal cost curve, or total cost, of X_2 , at the m^{th} quantity of X_2 .

$W_n = \int_0^{Y_n} (90 - 1.2Y) dY = 90Y - 0.6Y^2$, the area under the demand function (marginal revenue under competitive assumptions), or total revenue for Y , at the n^{th} quantity of Y .

a_m , the level at which the m^{th} quantity steps of X_2 is supplied in the solution

a_n , the level at which the n^{th} quantity step of Y is demanded in the solution.

a_p , the level at which the p^{th} step in the production of Y , from inputs X_1 and X_2 , enters the solution. Note that the index P embodies both ratios and quantities. Referring to Figure 2 may be of some help. Given a particular input ratio i , as quantities j of Y are changed, quantities of X_1 and X_2 change accordingly. Therefore the index p runs over both ratio numbers, and the quantity steps in Y , or $P = (\text{ratios})(M)$.

Linearization Parameters, Problem 1.

Supply of X_2 :

Total cost: $W = -0.5X_2^2$ area under supply

Initial $X_2 = 0$

$\Delta X_2 = 10$

STEPS = 12

Production of Y

$$Y = 4X_1^3 X_2^5$$

Initial Y = 10

$\Delta Y = 10$

STEPS = 5

RATIOS

R1 = .4 R2 = .8 R3 = 1.4 R4 = 1.8

Demand for Y

$W = 90Y - .6Y$ (area under demand)

Initial Y = 0

$\Delta Y = 7.5$

STEPS = 12

The matrix, data input, computer specification and results follows:

Table 2. Problem 1 Tableau Specification

No. 1 TEST TRAD. LP, INPUT, SUPPLY, PRODUCTION FUNCTION AND DEMAND									
	LACT 1	LACT 2	LACT 3	X_1	X_2	X_3	SUPPLY OF X_2	PRODUCTION	DEMAND, Y
OBJECTIVE	.5	.9	.7	-1.5	0	0	$\gamma_1 - \gamma_2 \dots - \gamma_2$	0 0	$W_1 \dots \dots W_n$
CONSTRAINT									
1.	.4	.3					Nonaugmented submatrix		
2.	.3		.2						
3.		.4	.9	1					
4.					1		$\gamma_1 - \gamma_2 \dots - \gamma_2$	Augmented submatrix	0
5.							1 1 1		1
6.				-1					0
7.					-1			$X_1 \dots \dots X_1$	0
8.								$X_2 \dots \dots X_2$	0
9.								$\gamma_1 \dots \dots \gamma_p$	0
10.								1 1	1
11.								$-Y_1 \dots \dots -Y_n$	0
								1 1	1

Problem One Input

```

1. 001      0      3
2. ** NO. 1 = TEST TRAD. LP. INPUT, SUPPLY, PRODUCTION FUNC AND DEMAND SPECIFICATION
3.  LACT1  LACT2  LACT3  X1T   X2T   YT
4.  1+0.5          2+0.9          3+0.7          4-1.5          -999
5.  1+0.4          2+0.3          -100+90.0
6.  1+0.5          3+0.2          -100+80.0
7.  2+0.4          3+0.9          4+1.0          -100+200.0
8.
9.  1
10. 5+1.0
11. 1-0.5          2.0
12. 0.0          10.0          12
13. 3          99999
14. 1
15. 4-1.0          -100+0.0
16. 1
17. 5-1.0          -100+0.0
18. 1
19. 6-1.0          -300+0.0
20. 40.4          0.8          1.4          1.8
21. 4.0          0.3          0.5
22. 10.0          10.0          5
23. 1
24. 6+1.0
25. 290.0          1.0          -0.6          2.0
26. 0.0          7.5          12
>
>
>

```

An explanation for each card follows

Problem One, Input Explanation

1. maximum problem, MPOS regular algorithm, 6 columns and 3 rows in the
nonaugmented matrix
2. title
3. variable names
4. objective, nonaugmented
5. row 1, coefficients by column, constraint type, RHS value
6. row 2, coefficients by column, constraint type, RHS value
7. row 3, coefficients by column, constraint type, RHS value
8. blank card for no data insertion
- 9-12 generate supply of X2
9. 1 nonzero coefficient in nonaugmented portion of row 4
10. entry of row 4, column 5, equal to 1.0.
11. integrated supply function, 1 term, coefficient = -0.5, exponent = 2.0
12. initial step = 0.0, increment = 10, 12 steps
- 13-22 generate production surface $Y = AX^{\alpha 1} X^{\alpha 2}$
13. 3 rows generated, 99999 = DEPV subroutine
14. 1 nonzero coefficient in row 6, (X1)
15. row 6, coefficient by column, constraint type, RHS value
16. 1 coefficient row 7, (X2)
17. row 7
18. 1 coefficient row 8, (Y)
19. row 8
20. 4 ratios, $\gamma_1 = 0.4$, $\gamma_2 = 0.8$, $\gamma_3 = 1.4$, $\gamma_4 = 1.8$
21. production function $Y = 4X_1^{0.3} X_2^{0.5}$
22. initial Y = 10, increment = 10, 5 steps
- 23-26 generate demand for Y
23. 1 coefficient row 10

- 24. row 10 coefficient
- 25. integrated demand function, 2 terms, coefficient 1 = 90, exponent
1 = 1.0, coefficient 2 = -0.6, exponent 2 = 2.0
- 26. initial Y = 0.0, increment = 7.5, 12 steps

Following is the computer output generated for problem one.

MP05 VERSION 4.0 NORTHWESTERN UNIVERSITY

```
*****  
*                                     *  
*                               M P O S                               *  
*                                     *  
*                               VERSION 4.0                           *  
*                                     *  
* MULTI-PURPOSE OPTIMIZATION SYSTEM                                *  
*                                     *  
*****
```

*** PROBLEM NUMBER 1 ***

```

REGULAR
TITLE
** NO. 1 # TEST TRAD. LP INPUT, SUPPLY, PRODUCTION FUND AND DEMAND SPECIFICATION AND
VARIABLES
    LACT1      LACT2      LACT3      X1T      X2T      Y1      Y2      Y3      Y4      Y5      Y6      Y7      Y8      Y9      Y10      Y11      Y12      Y13      Y14      Y15      Y16
    YA3      YA4      YA5      YA6      YA7      YA8      YA9      YB1      YB2      YB3      YB4      YB5      YB6      YB7      YB8      YB9      YB10      YB11      YB12      YB13      YB14
    YB15      YB16      YC1      YC2      YC3      YC4      YC5      YC6      YC7      YC8      YC9      YC10      YC11      YC12      YC13      YC14      YC15      YC16      YC17      YC18      YC19
    YC20      YC21      YC22      YC23      YC24      YC25      YC26      YC27      YC28      YC29      YC30      YC31      YC32      YC33      YC34      YC35      YC36      YC37      YC38      YC39      YC40
    YC41      YC42      YC43      YC44      YC45      YC46      YC47      YC48      YC49      YC50      YC51      YC52      YC53      YC54      YC55      YC56      YC57      YC58      YC59      YC60      YC61
    YC62      YC63      YC64      YC65      YC66      YC67      YC68      YC69      YC70      YC71      YC72      YC73      YC74      YC75      YC76      YC77      YC78      YC79      YC80      YC81      YC82
    YC83      YC84      YC85      YC86      YC87      YC88      YC89      YC90      YC91      YC92      YC93      YC94      YC95      YC96      YC97      YC98      YC99      YC100      YC101      YC102      YC103
    YC104      YC105      YC106      YC107      YC108      YC109      YC110      YC111      YC112      YC113      YC114      YC115      YC116      YC117      YC118      YC119      YC120      YC121      YC122      YC123      YC124
    YC125      YC126      YC127      YC128      YC129      YC130      YC131      YC132      YC133      YC134      YC135      YC136      YC137      YC138      YC139      YC140      YC141      YC142      YC143      YC144      YC145
    YC146      YC147      YC148      YC149      YC150      YC151      YC152      YC153      YC154      YC155      YC156      YC157      YC158      YC159      YC160      YC161      YC162      YC163      YC164      YC165      YC166
    YC167      YC168      YC169      YC170      YC171      YC172      YC173      YC174      YC175      YC176      YC177      YC178      YC179      YC180      YC181      YC182      YC183      YC184      YC185      YC186      YC187
    YC188      YC189      YC190      YC191      YC192      YC193      YC194      YC195      YC196      YC197      YC198      YC199      YC200      YC201      YC202      YC203      YC204      YC205      YC206      YC207      YC208
    YC209      YC210      YC211      YC212      YC213      YC214      YC215      YC216      YC217      YC218      YC219      YC220      YC221      YC222      YC223      YC224      YC225      YC226      YC227      YC228      YC229
    YC230      YC231      YC232      YC233      YC234      YC235      YC236      YC237      YC238      YC239      YC240      YC241      YC242      YC243      YC244      YC245      YC246      YC247      YC248      YC249      YC250
    YC251      YC252      YC253      YC254      YC255      YC256      YC257      YC258      YC259      YC260      YC261      YC262      YC263      YC264      YC265      YC266      YC267      YC268      YC269      YC270      YC271
    YC272      YC273      YC274      YC275      YC276      YC277      YC278      YC279      YC280      YC281      YC282      YC283      YC284      YC285      YC286      YC287      YC288      YC289      YC290      YC291      YC292
    YC293      YC294      YC295      YC296      YC297      YC298      YC299      YC300      YC301      YC302      YC303      YC304      YC305      YC306      YC307      YC308      YC309      YC310      YC311      YC312      YC313
    YC314      YC315      YC316      YC317      YC318      YC319      YC320      YC321      YC322      YC323      YC324      YC325      YC326      YC327      YC328      YC329      YC330      YC331      YC332      YC333      YC334
    YC335      YC336      YC337      YC338      YC339      YC340      YC341      YC342      YC343      YC344      YC345      YC346      YC347      YC348      YC349      YC350      YC351      YC352      YC353      YC354      YC355
    YC356      YC357      YC358      YC359      YC360      YC361      YC362      YC363      YC364      YC365      YC366      YC367      YC368      YC369      YC370      YC371      YC372      YC373      YC374      YC375      YC376
    YC377      YC378      YC379      YC380      YC381      YC382      YC383      YC384      YC385      YC386      YC387      YC388      YC389      YC390      YC391      YC392      YC393      YC394      YC395      YC396      YC397
    YC398      YC399      YC400      YC401      YC402      YC403      YC404      YC405      YC406      YC407      YC408      YC409      YC410      YC411      YC412      YC413      YC414      YC415      YC416      YC417      YC418
    YC419      YC420      YC421      YC422      YC423      YC424      YC425      YC426      YC427      YC428      YC429      YC430      YC431      YC432      YC433      YC434      YC435      YC436      YC437      YC438      YC439
    YC440      YC441      YC442      YC443      YC444      YC445      YC446      YC447      YC448      YC449      YC450      YC451      YC452      YC453      YC454      YC455      YC456      YC457      YC458      YC459      YC460
    YC461      YC462      YC463      YC464      YC465      YC466      YC467      YC468      YC469      YC470      YC471      YC472      YC473      YC474      YC475      YC476      YC477      YC478      YC479      YC480      YC481
    YC482      YC483      YC484      YC485      YC486      YC487      YC488      YC489      YC490      YC491      YC492      YC493      YC494      YC495      YC496      YC497      YC498      YC499      YC500      YC501      YC502
    YC503      YC504      YC505      YC506      YC507      YC508      YC509      YC510      YC511      YC512      YC513      YC514      YC515      YC516      YC517      YC518      YC519      YC520      YC521      YC522      YC523
    YC524      YC525      YC526      YC527      YC528      YC529      YC530      YC531      YC532      YC533      YC534      YC535      YC536      YC537      YC538      YC539      YC540      YC541      YC542      YC543      YC544
    YC545      YC546      YC547      YC548      YC549      YC550      YC551      YC552      YC553      YC554      YC555      YC556      YC557      YC558      YC559      YC560      YC561      YC562      YC563      YC564      YC565
    YC566      YC567      YC568      YC569      YC570      YC571      YC572      YC573      YC574      YC575      YC576      YC577      YC578      YC579      YC580      YC581      YC582      YC583      YC584      YC585      YC586
    YC587      YC588      YC589      YC590      YC591      YC592      YC593      YC594      YC595      YC596      YC597      YC598      YC599      YC600      YC601      YC602      YC603      YC604      YC605      YC606      YC607
    YC608      YC609      YC610      YC611      YC612      YC613      YC614      YC615      YC616      YC617      YC618      YC619      YC620      YC621      YC622      YC623      YC624      YC625      YC626      YC627      YC628
    YC629      YC630      YC631      YC632      YC633      YC634      YC635      YC636      YC637      YC638      YC639      YC640      YC641      YC642      YC643      YC644      YC645      YC646      YC647      YC648      YC649
    YC650      YC651      YC652      YC653      YC654      YC655      YC656      YC657      YC658      YC659      YC660      YC66
```

WPOS VERSION 4.0

NORTHWESTERN UNIVERSITY

* PROBLEM NUMBER 1 *

USING REGULAR

** NO. 1 TEST IRAD. LP INPUT, SUPPLY, PRODUCTION FUNC AND DEMAND SPECIFICATION *

6.	+.LE.	1.00000YA10	+	1.00000YA11	+	1.00000YA12
		1.00000				
	+	1.00000X1T	+			
	+	1.77300YB1	+	4.21700YB2	+	7.00000YB3
	+	10.03000YB4	+	13.25600YB5	+	2.73000YB6
	+	6.50300YB7	+	10.78600YB8	+	15.45000YB9
	+	20.44000YB10	+	3.07000YB11	+	9.20000YB12
	+	15.31600YB13	+	21.04500YB14	+	20.00000YB15
	+	4.53900YB16	+	10.70400YB17	+	17.02000YB18
	+	25.67700YB19	+	33.03800YB20		
7.	+.LE.		0			
	+	1.00000X2T	+			
	+	4.43300YB1	+	10.54200YB2	+	17.50100YB3
	+	25.07400YB4	+	33.14100YB5	+	13.41400YB6
	+	8.12900YB7	+	13.49500YB8	+	10.75000YB9
	+	25.55500YB10	+	2.77100YB11	+	6.55000YB12
	+	10.94000YB13	+	15.67500YB14	+	20.71000YB15
	+	2.52200YB16	+	5.09000YB17	+	0.05600YB18
	+	14.26500YB19	+	18.85400YB20		
8.	+.LE.		0			
	+	1.00000YT	+			
	+	10.00000YB1	+	20.00000YB2	+	30.00000YB3
	+	40.00000YB4	+	50.00000YB5	+	10.00000YB6
	+	20.00000YB7	+	30.00000YB8	+	40.00000YB9
	+	50.00000YB10	+	10.00000YB11	+	20.00000YB12
	+	30.00000YB13	+	40.00000YB14	+	50.00000YB15
	+	10.00000YB16	+	20.00000YB17	+	30.00000YB18
	+	40.00000YB19	+	50.00000YB20		
9.	+.GE.		0			
	+	1.00000YB1	+	1.00000YB2	+	1.00000YB3
	+	1.00000YB4	+	1.00000YB5	+	1.00000YB6
	+	1.00000YB7	+	1.00000YB8	+	1.00000YB9
	+	1.00000YB10	+	1.00000YB11	+	1.00000YB12
	+	1.00000YB13	+	1.00000YB14	+	1.00000YB15
	+	1.00000YB16	+	1.00000YB17	+	1.00000YB18
	+	1.00000YB19	+	1.00000YB20		
10.	+.LE.		1.00000			
	+	1.00000YT	+			
	+	0YC1	-	7.50000YC2	-	15.00000YC3
	+	22.50000YC4	-	30.00000YC5	-	37.50000YC6
	+	45.00000YC7	-	52.50000YC8	-	60.00000YC9
	+	67.50000YC10	-	75.00000YC11	-	82.50000YC12
11.	+.GE.		0			
	+	1.00000YC1	+	1.00000YC2	+	1.00000YC3
	+	1.00000YC4	+	1.00000YC5	+	1.00000YC6
	+	1.00000YC7	+	1.00000YC8	+	1.00000YC9
	+	1.00000YC10	+	1.00000YC11	+	1.00000YC12
	+.LE.		1.00000			
	OPTIMIZE					

MPQS VERSION 6.0

NORTHWESTERN UNIVERSITY

* PROBLEM NUMBER 1 *

USING REGULAR

NO. 1 N TEST TRAD. LP INPUT SUPPLY PRODUCTION FUNC AND DEMAND SPECIFICATION

SUMMARY OF RESULTS

VAR NO	VAR NAME	ROW NO	STATUS	ACTIVITY LEVEL	OPPORTUNITY COST	LOWER BOUND	UPPER BOUND
1	LACT1	1	LB	0.0000000	.2851852	0.0000	INF
2	LACT2	2	LB	308.0000000	0.0000000	0.0000	INF
3	LACT3	3	LB	51.1000000	0.0000000	0.0000	INF
4	XIT	4	LB	33.9300000	0.0000000	0.0000	INF
5	XIT	5	LB	13.8540000	0.0000000	0.0000	INF
6	Y1	6	LB	50.0000000	0.0600000	0.0000	INF
7	Y2	7	LB	0.0000000	100.0000000	0.0000	INF
8	Y3	8	LB	.1146000	0.0000000	0.0000	INF
9	Y4	9	LB	.8854000	0.0000000	0.0000	INF
10	Y5	10	LB	0.0000000	100.0000000	0.0000	INF
11	Y6	11	LB	0.0000000	300.0000000	0.0000	INF
12	Y7	12	LB	0.0000000	300.0000000	0.0000	INF
13	Y8	13	LB	0.0000000	1000.0000000	0.0000	INF
14	Y9	14	LB	0.0000000	1500.0000000	0.0000	INF
15	Y10	15	LB	0.0000000	2100.0000000	0.0000	INF
16	Y11	16	LB	0.0000000	2500.0000000	0.0000	INF
17	Y12	17	LB	0.0000000	3500.0000000	0.0000	INF
18	Y13	18	LB	0.0000000	4500.0000000	0.0000	INF
19	Y14	19	LB	0.0000000	5700.0000000	0.0000	INF
20	Y15	20	LB	0.0000000	7500.0000000	0.0000	INF
21	Y16	21	LB	0.0000000	5400.0000000	0.0000	INF
22	Y17	22	LB	0.0000000	3500.0000000	0.0000	INF
23	Y18	23	LB	0.0000000	167.1900000	0.0000	INF
24	Y19	24	LB	0.0000000	667.3000000	0.0000	INF
25	Y20	25	LB	0.0000000	721.6300000	0.0000	INF
26	Y21	26	LB	0.0000000	695.9000000	0.0000	INF
27	Y22	27	LB	0.0000000	280.1000000	0.0000	INF
28	Y23	28	LB	0.0000000	60.7700000	0.0000	INF
29	Y24	29	LB	0.0000000	550.0000000	0.0000	INF
30	Y25	30	LB	0.0000000	704.7000000	0.0000	INF
31	Y26	31	LB	0.0000000	450.0000000	0.0000	INF
32	Y27	32	LB	0.0000000	230.8000000	0.0000	INF
33	Y28	33	LB	0.0000000	15.7200000	0.0000	INF
34	Y29	34	LB	0.0000000	942.0000000	0.0000	INF
35	Y30	35	LB	0.0000000	690.4000000	0.0000	INF
36	Y31	36	LB	0.0000000	450.0000000	0.0000	INF
37	Y32	37	LB	0.0000000	227.3000000	0.0000	INF
38	Y33	38	LB	1.0000000	0.0000000	0.0000	INF
39	Y34	39	LB	0.0000000	1417.5000000	0.0000	INF
40	Y35	40	LB	0.0000000	1212.5000000	0.0000	INF
41	Y36	41	LB	0.0000000	675.0000000	0.0000	INF
42	Y37	42	LB	0.0000000	405.0000000	0.0000	INF
43	Y38	43	LB	0.0000000	202.5000000	0.0000	INF
44	Y39	44	LB	0.0000000	67.5000000	0.0000	INF
45	Y40	45	LB	.3333333	0.0000000	0.0000	INF

MPOS VERSION 4.0

NORTHWESTERN UNIVERSITY

* PROBLEM NUMBER 1 *

USING REGULAR

** NO. 1 # TEST TRAD. LP INPUT SUPPLY PRODUCTION FUNC AND DEMAND SPECIFICATION**

SUMMARY OF RESULTS

VAR NO	VAR NAME	RO#	STATUS	ACTIVITY LEVEL	OPPORTUNITY COST	LOWER BOUND	UPPER BOUND
46	YC8	---	B	0.6666667	0.0000000	0.00000	INF
47	YC9	---	L	0.0000000	67.5000000	0.00000	INF
48	YC10	---	L	0.0000000	202.5000000	0.00000	INF
49	YC11	---	L	0.0000000	405.0000000	0.00000	INF
50	YC12	---	L	0.0000000	675.0000000	0.00000	INF
51	SLACK---	0-	1	0.0000000	1.9629630	0.00000	INF
52	SLACK---	0-	2	69.7647000	0.0000000	0.00000	INF
53	SLACK---	0-	3	0.0000000	0.7777778	0.00000	INF
54	SLACK---	0-	4	0.0000000	15.0000000	0.00000	INF
55	SLACK---	0-	5	0.0000000	100.0000000	0.00000	INF
56	SLACK---	0-	6	0.0000000	2.2777778	0.00000	INF
57	SLACK---	0-	7	0.0000000	15.0000000	0.00000	INF
58	SLACK---	0-	8	0.0000000	31.5000000	0.00000	INF
59	ARTIF---	0-	9	0.0000000	-31.5000000	0.00000	INF
60	SLACK---	0-	10	0.0000000	1214.8067778	0.00000	INF
61	SLACK---	0-	11	0.0000000	-31.5000000	0.00000	INF
61	SLACK---	0-	11	0.0000000	1417.5000000	0.00000	INF

MAXIMUM VALUE OF THE OBJECTIVE FUNCTION = 3064.629000

CALCULATION TIME WAS .0930 SECONDS FOR 19 ITERATIONS.

DATA STORAGE MEMORY = 002037(OCTAL) TOTAL MEMORY = 050000(OCTAL)
TOTAL TIME FOR THIS PROBLEM WAS .709 SECONDS

Output Interpretation - Problem One

A brief description of the output follows. Those needing further explanation should refer to the MPOS manual.[2]

REGULAR - The particular MPOS algorithm requested by the user.

TITLE - Followed by user provided title

VARIABLES - The names of variables in the nonaugmented position of the matrix are given first, followed by the augmented or generated variables. Variables associated with the supply of X2 YA1 through YA12; production, YB1 to YB20; demand for Y, YC1 to YC12.

MAXIMIZE - The type of optimization requested for the objective function that follows. Note that the sign and objective value for each variable is provided.

CONSTRAINT - Followed by each row constraint in the problem.

Variables having zero coefficients are not listed. Note that constraint (4) is the commodity balance row for X2. The values at each step are the quantities associated with the total cost values in the objective function. Constraint (5) is the convexity constraint for X2.

Summary of Results

For each variable the following information is provided.

STATUS - Whether the variable is in the optimal basis at a zero or positive value. LB indicates zero; B positive value.

ACTIVITY LEVEL - The level or value a variable takes on in the optimal solution.

OPPORTUNITY COST - The cost in terms of a change in the objective function given a marginal increase in the particular variable.

This item is used synonymously with shadow price or dual.

LOWER, UPPER BOUNDS - The lower and upper limits of a variable within which the opportunity cost is unchanged.

Interpretation of the slack variables of the row constraints is the equivalent of finding the values of the dual problem.

The first three slack variables are associated with row constraints of the nonaugmented portions of the matrix. They are the commodity balance rows of fixed resources.

STATUS - Whether the slack variable is in the optimal basis at a zero or positive level. LB indicates zero; B a positive value. That is, SLACK takes on a positive value only when the resource is not totally exhausted.

OPPORTUNITY COST - The change in the objective function given an additional unit of the commodity constrained, or the value of an additional unit of the commodity.

The resource in row 2 has a positive value (69.76), indicating that it is not used up or is not a constraining resource. Since it is not constraining, its worth or value is zero as indicated in the opportunity cost column.

Additional units of the resources in rows 1 and 3 would be worth \$1.96 and \$0.78 respectively.

The interpretation of the Lagrangians or dual values of the rows of the augmented portion of the matrix was discussed in the theory review above. Again this value is given as the opportunity cost here. These values, taken from the computer output are listed below.

λ_4 - dual row 4 - price of X_2 -	15.00
λ_5 - dual row 5 - producer surplus X_2 -	100.00
λ_9 - dual row 9 - producer surplus Y -	1214.89
λ_{10} - dual row 10 - price of Y -	31.50
λ_{11} - dual row 11 - consumer surplus Y -	1417.50

The value of the objective function is 3064.61.

Problem Two

In problem two, surplus is maximized from two limited resources, which may be combined by two different production functions into a commodity facing a downward sloping demand.

The nonlinear programming specification follows:

$$\begin{aligned} \text{Max } Z = & 2.0X_{11} - 2.0X_{12} - 1.8X_{21} - 1.8X_{22} + \int_0^Y (90 - 1.2Y) dY \\ & + \lambda_1 (25 - X_{21} - X_{22}) \\ & + \lambda_2 (75 - X_{11} - X_{12}) \\ & + \lambda_3 (Y - 4X_{11}^{.3} X_{21}^{.5} - 3X_{12}^{.6} X_{22}^{.15}) \end{aligned}$$

Linearization Parameters

Production Y_1

$$\begin{aligned} Y_1 &= 4X_{11}^{.3} X_{21}^{.5} \\ \text{Initial } Y_1 &= 10 \\ \Delta Y_1 &= 10 \\ \text{STEPS} &= 5 \end{aligned}$$

RATIOS

$$R1 = .4 \quad R2 = .8 \quad R3 = 1.4 \quad R4 = 1.8$$

Production Y_2

$$\begin{aligned} Y_2 &= 3X_{12}^{.6} X_{22}^{.15} \\ \text{Initial } Y_2 &= 10 \\ \Delta Y_2 &= 10 \\ \text{STEPS} &= 5 \end{aligned}$$

Demand for Y

$$W = 90Y - .6Y^2 \quad \text{area under demand}$$

Initial Y = 0

$$\begin{aligned} \Delta Y &= 5 \\ \text{STEPS} &= 12 \end{aligned}$$

The matrix, data input, computer specification and results follow:

Table 3. Problem 2 Tableau Specification

NO. 2 TEST TWO PROD. FUNTS. AND ONE DEMAND									
	Y17	TV2	TX11	TX12	TX21	TX22	PROD 1	PROD 2	DEMAND
OBJECTIVE	0	0	-2.0	-2.0	-1.8	-1.8	0...0	0...0	$W_1 \dots W_n$
CONSTRAINT									
1.					1	1			< 25
2.			1	1					< 75
3.			-1				$X_{11} \dots X_{1n}$		< 0
4.					-1		$X_{21} \dots X_{2n}$		< 0
5.	-1						$Y_{11} \dots Y_{1n}$		> 0
6.							$1 \dots 1$		< 1
7.				-1				$X_{11} \dots X_{1n}$	< 0
8.						-1		$X_{21} \dots X_{2n}$	< 0
9.		-1						$Y_{21} \dots Y_{2n}$	> 0
10.								$1 \dots 1$	< 1
11.	1	1							$-Y_1 \dots -Y_n$
12.									$1 \dots 1$

Input Deck - Problem 2

```

1. 001      6      2
2. *** NO. 2 = TEST TWO PROD. ENTS. AND ONE DEMAND ***
3.      TY2      TY1      TX11      TX12      TX21      TX22
4.      3-2.0      -      4-2.0      -      5-1.8      -      6-1.8      -999
5.      5+1.0      -      6+1.0      -      -100+25.0
6.      3+1.0      -      4+1.0      -      -100+75.0
7.
8.      3      99999
9.      1
10.     3-1.0      -100+0.0
11.     1
12.     5-1.0      -100+0.0
13.     1
14.     2-1.0      -300+0.0
15.    40.4      0.8      1.4      1.8
16.    4.0      0.3      0.5
17.   10.0      10.0      5
18.     3      99999
19.     1
20.     4-1.0      -100+0.0
21.     1
22.     6-1.0      -100+0.0
23.     1
24.     1-1.0      -300+0.0
25.    4+0.4      +0.8      1.4      1.8
26.    3.0      0.6      0.15
27.   10.0      10.0      5
28.     2
29.     1+1.0      -      2+1.0
30.     2+90.0      1.0      -0.6      2.0
31.    0.0      5.0      12
>

```

Problem Two Input Explanation

1. maximum, MPOS regular algorithm, 6 columns and 2 rows in the nonaugmented matrix.
2. title
3. variable names
4. objective, nonaugmented
5. row 1, coefficients by column, constraint type, RHS value
6. row 2, coefficients by column, constraint type, RHS value
7. blank card for no data insertion
- 8-17 generate production surface $Y_1 = 4X_{11}^{.3} X_{21}^{.5}$
8. 3 rows generated, 99999 = DEPV subroutine
9. 1 nonzero coefficient in row 3, (XII)
10. row 3, coefficient by column, constraint type, RHS value
11. 1 nonzero coefficient in row 4, (X_{21})
12. row 4, coefficient by column, constraint type, RHS value
13. 1 nonzero coefficient in row 5, (Y_1)
14. row 5, coefficient by column, constraint type, RHS value
15. 4 ratios,
16. production function $Y_1 = 3X_{11}^{0.3} X_{21}^{0.5}$
17. initial Y=10, increment=10, 5 steps
- 18-27 generate production surface $Y_2 = 3X_{12}^{0.6} X_{22}^{0.15}$ similar to 8-17
- 28-31 generate demand for Y
28. 2 nonzero coefficients in row 11 (Y_1 and Y_2)
29. row 11, coefficients by column
30. integrated demand function, 2 terms, coefficient 1 = 90, exponent 1 = 1.0, coefficient 2 = -0.6, exponent 2 = 2.0
31. initial Y = 0.0, increment = 5.0, 18 steps

MPOS VERSION 4.0

NORTHWESTERN UNIVERSITY

```

*****
*                                     *
*               M P O S               *
*                                     *
*               VERSION 4.0           *
*                                     *
*  MULTI-PURPOSE OPTIMIZATION SYSTEM *
*                                     *
*****

```

**** PROBLEM NUMBER 1 ****

REGULAR

TITLE

*** NO. 2 # TEST TWO PROD. FNTS. AND ONE DEMAND ***

VARIABLES

TX2	TX1	TX11	TX12	TX21	TX22	YA1	YA2
YA3	YA4	YA5	YA6	YA7	YA8	YA9	YA10
YA11	YA12	YA13	YA14	YA15	YA16	YA17	YA18
YA19	YA20	YB1	YB2	YB3	YB4	YB5	YB6
YB7	YB8	YB9	YB10	YB11	YB12	YB13	YB14
YB15	YB16	YB17	YB18	YB19	YB20	YC1	YC2
YC3	YC4	YC5	YC6	YC7	YC8	YC9	YC10
YC11	YC12	YC13	YC14	YC15	YC16	YC17	YC18

MAXIMIZE

2.00000TX11	2.00000TX12	1.80000TX21
1.80000TX22		

0YA1	0YA2	0YA3
0YA4	0YA5	0YA6
0YA7	0YA8	0YA9
0YA10	0YA11	0YA12
0YA13	0YA14	0YA15
0YA16	0YA17	0YA18
0YA19	0YA20	0YB1
0YB2	0YB3	0YB4
0YB5	0YB6	0YB7
0YB8	0YB9	0YB10
0YB11	0YB12	0YB13
0YB14	0YB15	0YB16
0YB17	0YB18	0YB19
0YB20	0YC1	435.00000YC2
840.00000YC3	1215.00000YC4	1560.00000YC5
1875.00000YC6	2160.00000YC7	2415.00000YC8
2640.00000YC9	2835.00000YC10	3000.00000YC11
3135.00000YC12	3240.00000YC13	3315.00000YC14
3360.00000YC15	3375.00000YC16	3360.00000YC17
3315.00000YC18		

CONSTRAINT

1.	+	1.00000TX21	+	1.00000TX22	
	.Le.	+	25.00000		
2.	+	1.00000TX11	+	1.00000TX12	
	.Le.	+	75.00000		
3.	-	1.00000TX11			
	+	1.77300YA1	+	4.21700YA2	7.00000YA3
	+	10.03000YA4	+	13.25600YA5	2.73400YA6
	+	6.50300YA7	+	10.79600YA8	15.46800YA9
	+	20.44400YA10	+	3.87900YA11	9.22700YA12

MPUS VERSION 4.0

NORTHWESTERN UNIVERSITY

* PROBLEM NUMBER 1 *

USING REGULAR

*** NO. 2: TEST TWO PROD. ENTS. AND ONE DEMAND ***

+	15.31600YA13	+	21.94500YA14	+	29.00500YA15
+	4.53900YA16	+	10.79600YA17	+	17.92100YA18
+	25.67700YA19	+	33.93800YA20		
.LE.	+	0			
4.	1.00000TX21				
+	4.43300YA1	+	10.54200YA2	+	17.50100YA3
+	25.07400YA4	+	33.14100YA5	+	3.41800YA6
+	8.12900YA7	+	13.49500YA8	+	19.33500YA9
+	25.55500YA10	+	2.77100YA11	+	6.59000YA12
+	10.94000YA13	+	15.67500YA14	+	20.71800YA15
+	2.52200YA16	+	5.99600YA17	+	9.95600YA18
+	14.26500YA19	+	18.85400YA20		
.LE.	+	0			
5.	1.00000TY1				
+	10.00000YA1	+	20.00000YA2	+	30.00000YA3
+	40.00000YA4	+	50.00000YA5	+	10.00000YA6
+	20.00000YA7	+	30.00000YA8	+	40.00000YA9
+	50.00000YA10	+	10.00000YA11	+	20.00000YA12
+	30.00000YA13	+	40.00000YA14	+	50.00000YA15
+	10.00000YA16	+	20.00000YA17	+	30.00000YA18
+	40.00000YA19	+	50.00000YA20		
.LE.	+	0			
6.	1.00000YA1	+	1.00000YA2	+	1.00000YA3
+	1.00000YA4	+	1.00000YA5	+	1.00000YA6
+	1.00000YA7	+	1.00000YA8	+	1.00000YA9
+	1.00000YA10	+	1.00000YA11	+	1.00000YA12
+	1.00000YA13	+	1.00000YA14	+	1.00000YA15
+	1.00000YA16	+	1.00000YA17	+	1.00000YA18
+	1.00000YA19	+	1.00000YA20		
.LE.	+	1.00000			
7.	1.00000FX12				
+	4.14600YB1	+	10.44600YB2	+	17.93700YB3
+	26.32300YB4	+	35.44400YB5	+	4.76200YB6
+	11.99900YB7	+	20.60400YB8	+	30.23700YB9
+	40.71500YB10	+	5.32600YB11	+	13.42100YB12
+	23.04400YB13	+	33.81800YB14	+	45.53600YB15
+	5.80000YB16	+	14.11200YB17	+	24.23200YB18
+	35.58100YB19	+	47.88400YB20		
.LE.	+	0			
8.	1.00000TX22				
+	10.36400YB1	+	26.11500YB2	+	44.84200YB3
+	65.80700YB4	+	88.61000YB5	+	5.95300YB6
+	14.99900YB7	+	25.75500YB8	+	37.79600YB9
+	50.89300YB10	+	3.80400YB11	+	9.58600YB12
+	10.46000YB13	+	24.15500YB14	+	32.52600YB15
+	3.11100YB16	+	7.84000YB17	+	13.46200YB18
+	19.75000YB19	+	26.60200YB20		
.LE.	+	0			
9.	1.00000TY2				
+	10.00000YB1	+	20.00000YB2	+	30.00000YB3
+	40.00000YB4	+	50.00000YB5	+	10.00000YB6

MPOS VERSION 4.0

NORTHWESTERN UNIVERSITY

* PROBLEM NUMBER 1 *

USING REGULAR

*** NO. 2 # TEST TWO PROD. FUNTS. AND ONE DEMAND ***

10.	+	20.00000YB7	+	30.00000YB8	+	40.00000YB9
	+	50.00000YB10	+	10.00000YB11	+	20.00000YB12
	+	30.00000YB13	+	40.00000YB14	+	50.00000YB15
	+	10.00000YB16	+	20.00000YB17	+	30.00000YB18
	+	40.00000YB19	+	50.00000YB20		
	.GE.	+	0			
10.	+	1.00000YB1	+	1.00000YB2	+	1.00000YB3
	+	1.00000YB4	+	1.00000YB5	+	1.00000YB6
	+	1.00000YB7	+	1.00000YB8	+	1.00000YB9
	+	1.00000YB10	+	1.00000YB11	+	1.00000YB12
	+	1.00000YB13	+	1.00000YB14	+	1.00000YB15
	+	1.00000YB16	+	1.00000YB17	+	1.00000YB18
	+	1.00000YB19	+	1.00000YB20		
	.LE.	+	1.00000			
11.	+	1.00000Y2	+	1.00000Y1		
	-	0Y1	-	5.00000YC2	-	10.00000YC3
	-	15.00000YC4	-	20.00000YC5	-	25.00000YC6
	-	30.00000YC7	-	35.00000YC8	-	40.00000YC9
	-	45.00000YC10	-	50.00000YC11	-	55.00000YC12
	-	60.00000YC13	-	65.00000YC14	-	70.00000YC15
	-	75.00000YC16	-	80.00000YC17	-	85.00000YC18
	.GE.	+	0			
12.	+	1.00000YC1	+	1.00000YC2	+	1.00000YC3
	+	1.00000YC4	+	1.00000YC5	+	1.00000YC6
	+	1.00000YC7	+	1.00000YC8	+	1.00000YC9
	+	1.00000YC10	+	1.00000YC11	+	1.00000YC12
	+	1.00000YC13	+	1.00000YC14	+	1.00000YC15
	+	1.00000YC16	+	1.00000YC17	+	1.00000YC18
	.LE.	+	1.00000			
	OPTIMIZE					

MPQS VERSION 4.0

NORTHWESTERN UNIVERSITY

* PROBLEM NUMBER 1 *

USING REGULAR

*** NO. 2 A TEST TWO PROD. ENTS. AND ONE DEMAND ***

SUMMARY OF RESULTS

VAR	VAR	ROW	STATUS	ACTIVITY	OPPORTUNITY	LOWER	UPPER
NO	NAME	NO		LEVEL	COST	BOUND	BOUND
1	Y12	--	B	16.4178473	0.0000000	0.0000	INF
2	Y11	--	B	50.0000000	0.0000000	0.0000	INF
3	Y111	--	B	33.9380000	0.0000000	0.0000	INF
4	Y112	--	B	11.0628710	0.0000000	0.0000	INF
5	Y121	--	B	18.8540000	0.0000000	0.0000	INF
6	Y122	--	B	6.1480000	0.0000000	0.0000	INF
7	Y11	--	LB	0.0000000	73.1310074	0.0000	INF
8	Y12	--	LB	0.0000000	82.2906043	0.0000	INF
9	Y13	--	LB	0.0000000	105.2450556	0.0000	INF
10	Y14	--	LB	0.0000000	138.1685041	0.0000	INF
11	Y15	--	LB	0.0000000	179.1071592	0.0000	INF
12	Y16	--	LB	0.0000000	59.3896412	0.0000	INF
13	Y17	--	LB	0.0000000	49.6261726	0.0000	INF
14	Y18	--	LB	0.0000000	51.0180968	0.0000	INF
15	Y19	--	LB	0.0000000	60.4825959	0.0000	INF
16	Y110	--	LB	0.0000000	76.4191000	0.0000	INF
17	Y111	--	LB	0.0000000	51.6957010	0.0000	INF
18	Y112	--	LB	0.0000000	31.3249520	0.0000	INF
19	Y113	--	LB	0.0000000	20.6303785	0.0000	INF
20	Y114	--	LB	0.0000000	16.9569681	0.0000	INF
21	Y115	--	LB	0.0000000	18.8984881	0.0000	INF
22	Y116	--	LB	0.0000000	49.1782345	0.0000	INF
23	Y117	--	LB	0.0000000	25.3274494	0.0000	INF
24	Y118	--	LB	0.0000000	10.6556917	0.0000	INF
25	Y119	--	LB	0.0000000	2.6524229	0.0000	INF
26	Y120	--	B	1.0000000	0.0000000	0.0000	INF
27	Y11	--	LB	0.0000000	109.0173390	0.0000	INF
28	Y12	--	LB	0.0000000	274.6803493	0.0000	INF
29	Y13	--	LB	0.0000000	488.6497784	0.0000	INF
30	Y14	--	LB	0.0000000	758.9451110	0.0000	INF
31	Y15	--	LB	0.0000000	1019.0737103	0.0000	INF
32	Y16	--	LB	0.0000000	42.1805853	0.0000	INF
33	Y17	--	LB	0.0000000	106.2487693	0.0000	INF
34	Y18	--	LB	0.0000000	199.4409761	0.0000	INF
35	Y19	--	LB	0.0000000	314.5167790	0.0000	INF
36	Y110	--	LB	0.0000000	447.5823434	0.0000	INF
37	Y111	--	LB	0.0000000	10.1460935	0.0000	INF
38	Y112	--	LB	0.0000000	25.5615602	0.0000	INF
39	Y113	--	LB	0.0000000	60.8843257	0.0000	INF
40	Y114	--	LB	0.0000000	111.1784284	0.0000	INF
41	Y115	--	LB	0.0000000	173.7922876	0.0000	INF
42	Y116	--	B	.3582153	0.0000000	0.0000	INF
43	Y117	--	B	.6417847	0.0000000	0.0000	INF
44	Y118	--	LB	0.0000000	16.9904119	0.0000	INF
45	Y119	--	LB	0.0000000	46.7808539	0.0000	INF

MIPOS VERSION 4.0

NORTHWESTERN UNIVERSITY

* PROBLEM NUMBER 1 *

USING REGULAR

*** NO. 2 " TEST TWO PROD. FUNTS. AND ONE DEMAND ***

SUMMARY OF RESULTS

VAR NO	VAR NAME	ROW NO	STATUS	ACTIVITY LEVEL	OPPORTUNITY COST	LOWER BOUND	UPPER BOUND
46	YB20	--	LB	0.0000000	87.0715348	0.0000	INF
47	YC1	--	LB	0.0000000	2750.0000000	0.0000	INF
48	YC2	--	LB	0.0000000	2340.0000000	0.0000	INF
49	YC3	--	LB	0.0000000	1960.0000000	0.0000	INF
50	YC4	--	LB	0.0000000	1650.0000000	0.0000	INF
51	YC5	--	LB	0.0000000	1350.0000000	0.0000	INF
52	YC6	--	LB	0.0000000	1060.0000000	0.0000	INF
53	YC7	--	LB	0.0000000	840.0000000	0.0000	INF
54	YC8	--	LB	0.0000000	650.0000000	0.0000	INF
55	YC9	--	LB	0.0000000	450.0000000	0.0000	INF
56	YC10	--	LB	0.0000000	300.0000000	0.0000	INF
57	YC11	--	LB	0.0000000	180.0000000	0.0000	INF
58	YC12	--	LB	0.0000000	90.0000000	0.0000	INF
59	YC13	--	LB	0.0000000	50.0000000	0.0000	INF
60	YC14	--	B	.7164305	0.0000000	0.0000	INF
61	YC15	--	B	.2835695	0.0000000	0.0000	INF
62	YC16	--	LB	0.0000000	50.0000000	0.0000	INF
63	YC17	--	LB	0.0000000	90.0000000	0.0000	INF
64	YC18	--	LB	0.0000000	180.0000000	0.0000	INF
65	SLACK--	1	LB	0.0000000	13.6315923	0.0000	INF
66	SLACK--	2	B	29.9991284	0.0000000	0.0000	INF
67	SLACK--	3	LB	0.0000000	2.0000000	0.0000	INF
68	SLACK--	4	LB	0.0000000	15.4315923	0.0000	INF
69	SLACK--	5	LB	0.0000000	9.0000000	0.0000	INF
70	ARTIF--	5	LB	0.0000000	-9.0000000	0.0000	INF
71	SLACK--	6	LB	0.0000000	91.1767587	0.0000	INF
72	SLACK--	7	LB	0.0000000	2.0000000	0.0000	INF
73	SLACK--	8	LB	0.0000000	15.4315923	0.0000	INF
74	SLACK--	9	LB	0.0000000	-9.0000000	0.0000	INF
75	ARTIF--	9	LB	0.0000000	-9.0000000	0.0000	INF
76	SLACK--	10	LB	0.0000000	50.7923163	0.0000	INF
77	SLACK--	11	LB	0.0000000	9.0000000	0.0000	INF
78	ARTIF--	11	LB	0.0000000	-9.0000000	0.0000	INF
79	SLACK--	12	LB	0.0000000	2750.0000000	0.0000	INF

MAXIMUM VALUE OF THE OBJECTIVE FUNCTION = 3192.756883

CALCULATION TIME WAS .2200 SECONDS FOR 30 ITERATIONS.

DATA STORAGE MEMORY = 002465 (OCTAL) TOTAL MEMORY = 050000 (OCTAL)
TOTAL TIME FOR THIS PROBLEM WAS .991 SECONDS

Problem Two Dual

For problem 2, the dual values are taken from the computer output and listed below:

λ_6 - dual row 6 - producer surplus Y_1 -	91.18
λ_{10} - dual row 10 - producer surplus Y_2 -	30.79
λ_{11} - dual row 11 - price of Y -	9.00
λ_{12} - dual row 12 - consumer surplus Y -	2730.00

VI. HEADER CARDS

Three groups of JCL header cards are given. The first is to generate maximum output for troubleshooting. The second is to utilize a previously compiled program and to generate only output pertinent to problem solving. The third is to address APEX-1.

1. Maximum output, No LP output

NAME,T10.

ACCOUNT,GQM1111,PASSWORD.

BIN CARD IF NECESSARY

RFL(77000)

ACQUIRE(MGAO).

FETCH(MINNLIB/V=MNF)

MNF(I=MGAO,B=CMG)

RETAIN,CMG/CT=PU.

CBR(INPUT,TAPE5)

R,TAPE5.

SETTL(20)

CMG.

COST.

⁷₈₉

Data Deck

⁷₈₉

⁶₇₈₉

⁶₇₈₉

2. Reduced output using compiled deck, CMG, with LP output

NAME,T10.

ACCOUNT,GQM1111,PASSWORD.

RFL(77000)

FETCH(MINNLIB/V-MNF)

ACQUIRE,CMG.

CBR(INPUT,TAPE5)

R,TAPE5.

SETTL(20)

CMG.

R,TAPE1.

COPYSBF,TAPE1,OUTPUT.

R,TAPE1.

MPOS(TAPE1)

COST.

⁷_{8₉}

Date Deck

⁷_{8₉}

⁶_{7_{8₉}}

⁶_{7_{8₉}}

3. APEX-1

In order to access APEX, the following cards are inserted between the MPOS(TAPE1) and COST cards in deck 2.

R,APXFIL.

COPYSBF(APXFIL,OUTPUT)

RETURN,TAPE1.

R,APXFIL.

RENAME,TAPE1=APXFIL.

APEX(SOLV-----)

VII. FINAL COMMENTS

This computer program has not been fully tested and problems may be found. However, initial testing and use indicate that the program could be extremely useful in solving certain types of problems. The fortran program itself is rather straight forward and appears to be organized in a way that would facilitate user provided modification.

Some comments regarding the use of MGAO with APEX are in order. Currently, only APEX-I one is operable. APEX-II may be accessed, however format errors are incurred. If the user accesses APEX-I directly, parametric and/or ranging procedures are difficult if not impossible to perform. An alternative is to use MGAO to punch out the data deck and then put together a completely new problem specification for the APEX problem. This may be done with cards or through the use of permanent files and interactive terminals.

The user should not be limited by the mathematical forms of the functions as currently read. One need only to change the format and the read statements to fit different needs and problems.

Bibliography

- [1] Program developed by Terry L. Roe and Paul Chang
- [2] Multi Purpose Optimization System User's Guide Version 3, Claude Cohen and Jack Stein, Manual No. 320, Copyright 1976, Voeglback Computing Center, Northwestern University, Evanston, Illinois 60201.

APEX-1 Reference Manual

- [3] Intrilligator Michael D., Mathematical Optimization and Economics Theory, 1971, Prentice-Hall, Inc., p. 72.
- [4] Intrilligator, p.44.

- [5] Duloy, John H. and Roger D. Norton, "Prices and Incomes in Linear Programming Models," American Journal of Agricultural Economics, Volume 57, Number 4 (November 1975) pp. 591-600.

Klein, Harold E. and Terry L. Roe, "Agriculture Sector Analysis Model Design: The Influence of Administrative Infrastructure Characteristics," in Planning Processes in Developing Countries: Techniques and Achievements, eds. W.D. Cook and T.E. Kuhn (Amsterdam-London: North-Holland, 1982) pp. 273-308.

- [6] Klein and Roe, pp. 297-299.
- [7] Duloy, John H. and Roger P. Norton, "The CHAC Demand Structures, Chapter 3 in Programming Studies for Mexican Agricultural Policy, eds. Roger D. Norton and Leopoldo M. Solis, forthcoming.
- [8] Roe, Terry, "Modelling of Nonlinear Functions into a Linear Programming Format," Staff Paper P75-9, June 1975, Dept. of Ag. and Applied Econ., U of M, St. Paul.

Appendix A. Variable Name

```

*****
** METHOD(M1) CONTAINS THE LP ALGORITHMS WHICH ARE USED IN M.P.O.S. *
** M1 = 01 #REGULAR#, 2-PHASE SIMPLEX (L.P.) *
** M1 = 02 #REVISED#, REVISED SIMPLEX (L.P.) *
** M1 = 03 #DUAL#, DUAL SIMPLEX (L.P.) *
** M1 = 04 #MINIT#, PRIMAL-DUAL ALG. (L.P.) *
** M1 = 05 #BBMIP#, BRANCH AND BOUND MIXED INT. PROGRAM.(I.P.) *
** M1 = 06 #DSZLIP#, DIRECT SEARCH 0-1 INTEGER PROGRAM. (I.P.) *
** M1 = 07 #GOMORY#, GOMORY'S CUTTING PLANE (I.P.) *
** M1 = 08 #WOLFE#, WOLFE'S QUADRATIC SIMPLEX (Q.P.) *
** M1 = 09 #REALP#, BEALE'S ALGORITHM (Q.P.) *
** M1 = 10 #LEMKE#, LEMKE'S COMPLEMENTARY PIVOT ALG.(Q.P.) *
** M1 = 11 #APEX1#, MPOS-APEX DATA FILE INTERFACE (GENERAL) *
** M1 = 12 #APEX2#, MPOS-APEX DATA FILE INTERFACE (GENERAL) *
**
** VARS(N1) ARE THE RESERVED WORDS USED BY #MPOS#. *
** N1 = 1 TITLE *
** N1 = 2 INTEGER *
** N1 = 3 VARIABLES *
** N1 = 4 MAXIMIZE *
** N1 = 5 MINIMIZE *
** N1 = 6 CONSTRAINTS *
** N1 = 7 BOUNDS *
** N1 = 8 PRINT *
** N1 = 9 OPTIMIZE *
** N1 = 10 ENDAPFX *
** N1 = 11 BNDALL *
** N1 = 12 BNDINT *
** N1 = 13 RNGOBJ *
** N1 = 14 RNGRHS *
** N1 = 15 TOLERANCE *
** N1 = 16 EPSILON *
** N1 = 17 BNDOBJ *
** N1 = 18 LIMIT *
** N1 = 19 NOSCALE *
** N1 = 20 CHECK *
** N1 = 21 QCHECK *
** N1 = 22 GO *
** N1 = 23 STOP *
** N1 = 24 RESCALE *
** N1 = 25 MAXCM *
** N1 = 26 MAXECs *
**
** GFL(M2) CONTAIN #.LE.#, #.EQ.#, #.GE.# *
** ==-100 MEANS LESS THAN OR EQUAL TO *
** ==-200 MEANS EQUAL TO *
** ==-300 MEANS GREAT THAN OR EQUAL TO *
**

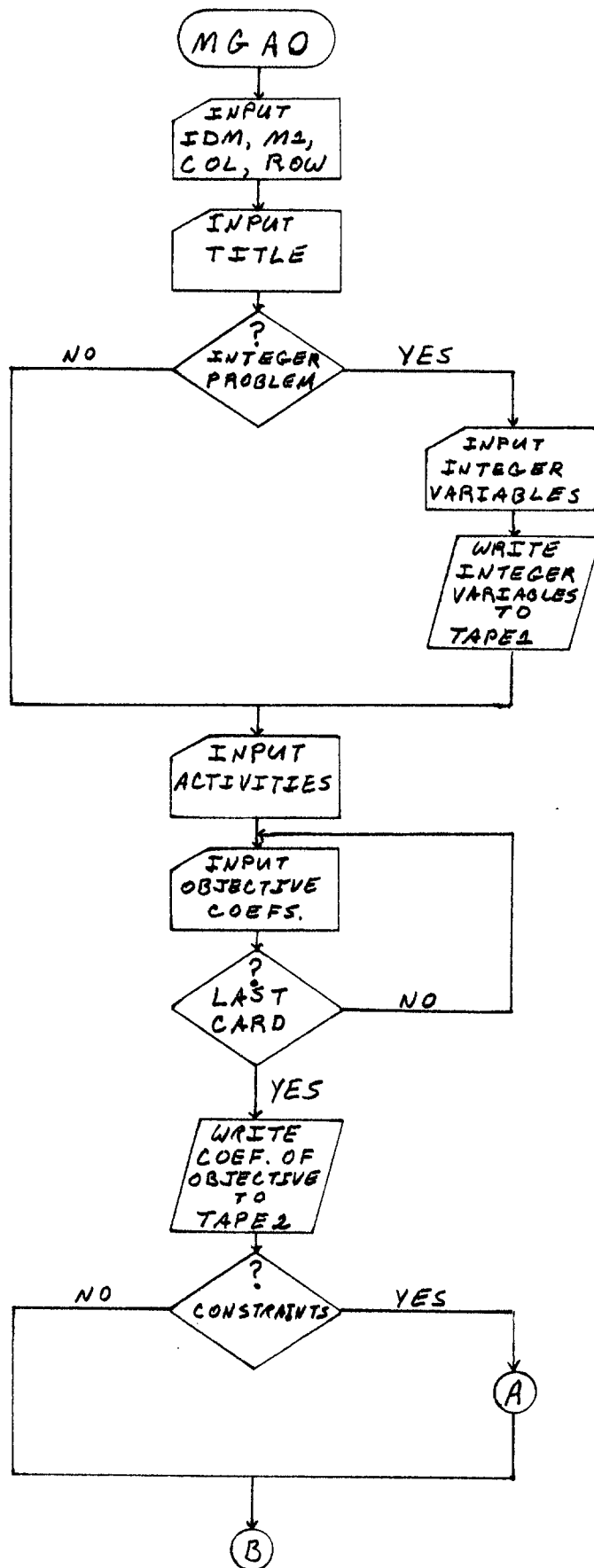
```

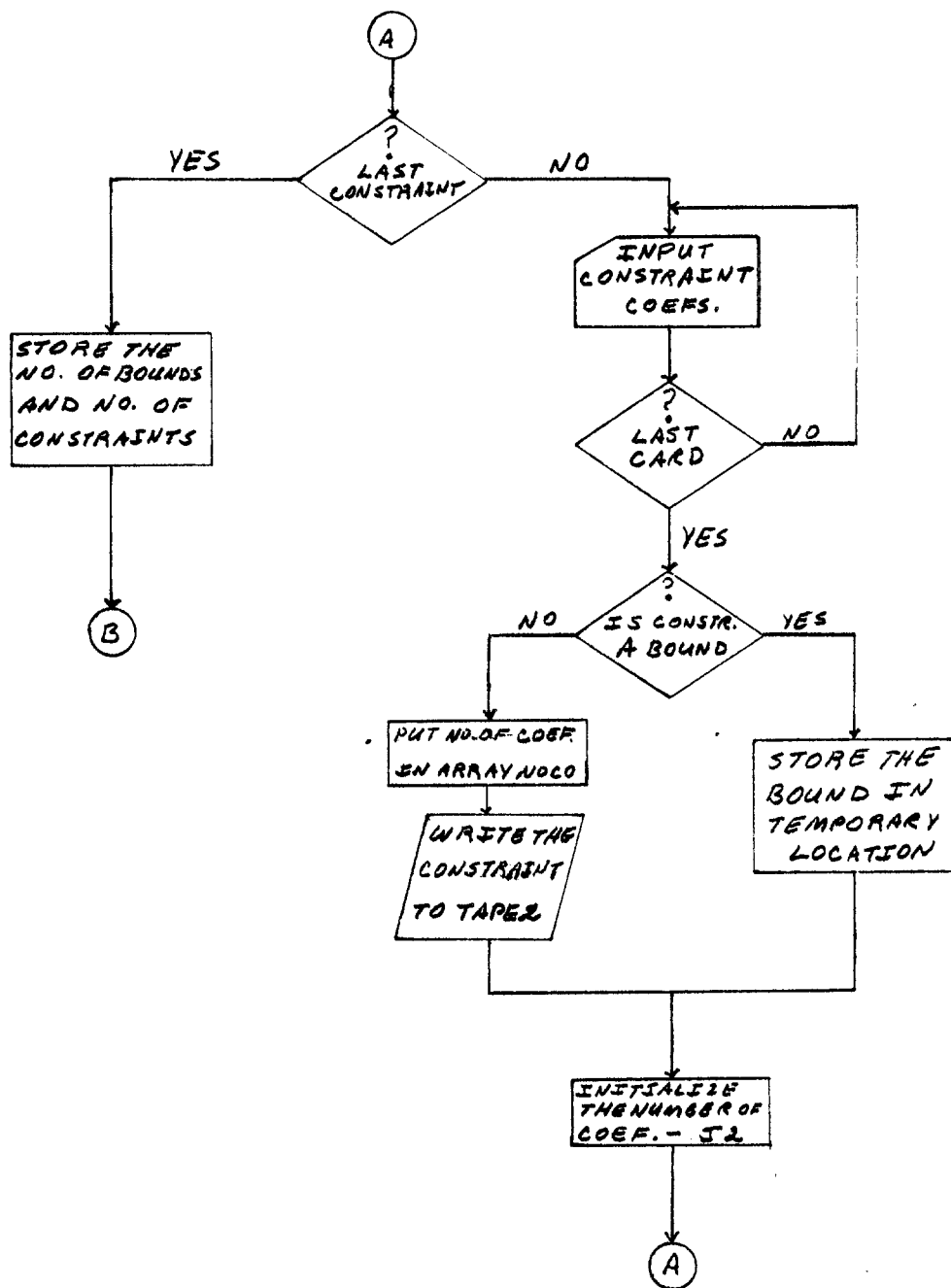
```

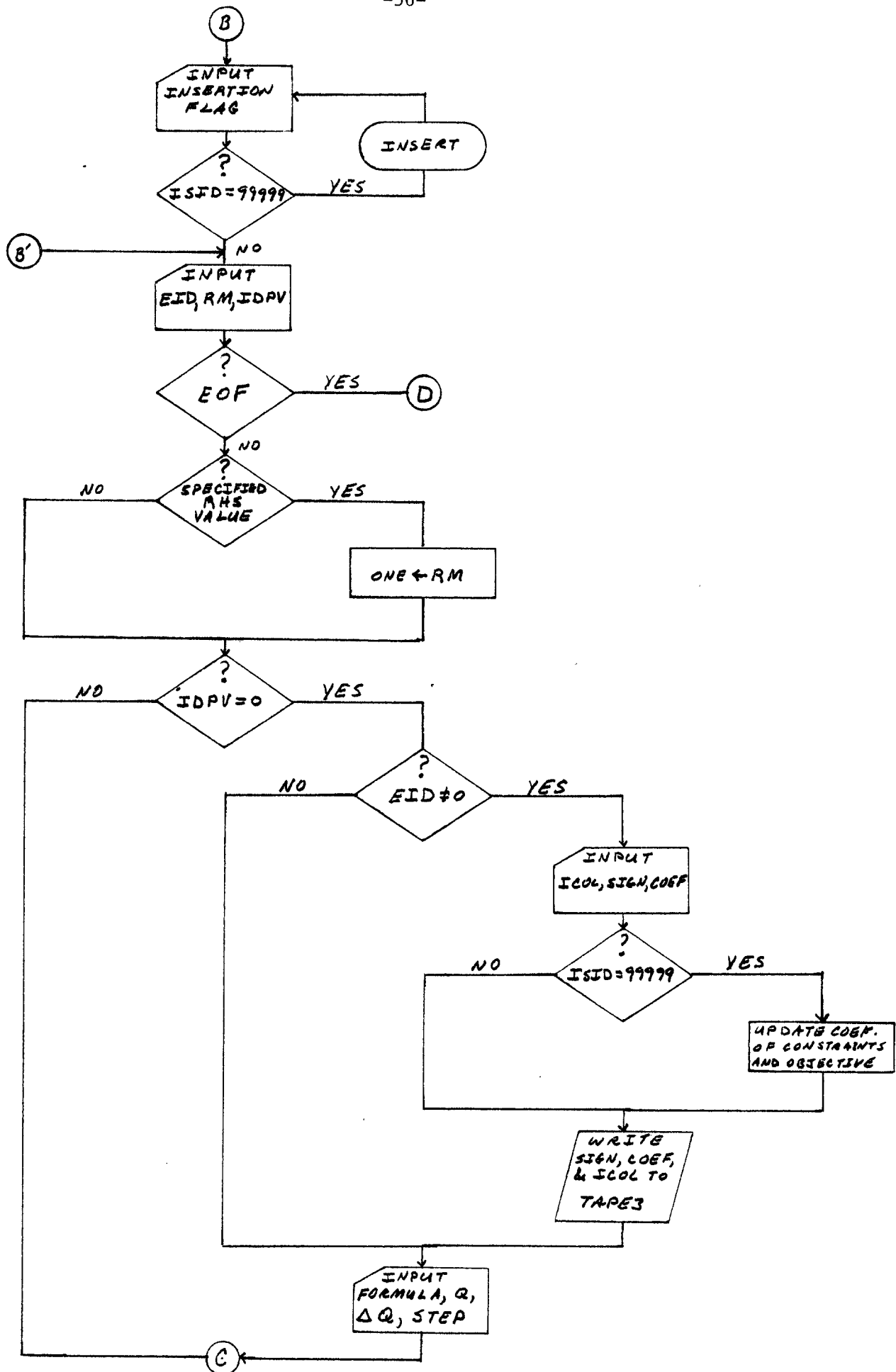
** CWRITE      AN ARRAY STORE THE COEF. OF MATRIX TO BE DUMPED
** EID          EXTENDING MODEL FLAG
**             .EQ. 0      (ONLY ADD ONE BLANK CARD)
**             .NE. 0      EXTENDING THE MODEL WITH READING THE COEF.
**             (PUNCH THE NO. OF COEF. WHICH WILL BE ADDED.)
** IDM          MAX. OR MIN. IDENTIFIER
**             = 0      MEANS MAXIMIZE
**             = 1      MEANS MINIMIZE
** IDPV          2 INDEP. VAR. FUNC. EXTENSION
**             = 0      OTHER CASE
** ISID          INSERTION FLAG
**             = 0      NO NEW ACTIVITIES ARE TO BE ADDED.
**             = 99999  NEW ACTIVITIES ARE TO BE INSERTED.
** IWRITE       DUMP MATRIX TABLE FLAG
**             = 0      DON'T DUMP THE TABLE
**             = 1      DUMP THE TABLE
**             = 2      DUMP PART OF THE TABLE
** ACT          ACTIVITIES
** AACT         NEW ACTIVITIES
** CC           COEF. OF THE FORMULA
** COEF         ABSOLUTE VALUE OF THE COEFFICIENT.
** COL          COLUMN NO. OF THE MATRIX(NO. OF ENTRIES)
** DELTAQ       DELTA Q
** ICOL         COLUMN COORD. OF THE ACTIVITY
** IEXP         EXPONENT OF EACH ITEM OF THE FORMULA
** IJ           NO. OF ITEMS OF THE FORMULA.
** JA           NO. OF EXTENDING PROCEDURES.
** N2           NO. OF INTEGER VARIABLES FOR I.P.
** NAA          NO. OF ACTIVITIES NEEDS TO BE INSERTED
** NBB          TOTAL NO. OF NON-ZERO COEF.
** NINS         THE PLACEMENT OF NEW ACTIVITIES WILL BE INSERTED
** NCOL         ARRAY STORE THE VALUES OF COL OF EXTENDING
** NEID         ARRAY STORE THE VALUES OF EID OF EXTENDING
** NOBU         NO. OF BOUNDS
** NOCD         NO. OF CONSTRAINTS
** NOCO         ARRAY STORE THE VALUES OF NOCO OF EXTENDING
** NROW         ARRAY STORE THE VALUES OF ROW OF EXTENDING
** NSTEP        ARRAY STORE THE VALUES OF STEP OF EXTENDING
** ONE          #1#, CONSTANT ONE.
** Q(1)         INITIAL VALUE OF Q
** ROW          ROW NO. OF THE MATRIX
** RM           THE SPECIFIED R.H.S VALUE DIFFERENT FROM DEFAULT ONE
** SIGN         SIGN OF THE ACTIVITY
** STEP         STEPS
** TACT         TEMP. LOC. STORES THE ACTIVITIES OF BOUNDS
** TCOEF        TEMP. LOC. STORES THE ABSOLUTE VALUES OF BOUNDS
** TITLE        TITLE OF THE PROBLEM (RESTRICTED ONE CARD )
** TSIGN        TEMP. LOC. STORES THE SIGN OF BOUNDS
** IT1          TEMP. LOC. STORES THE RELATIONS OF BOUNDS
** IT2          TEMP. LOC. STORES THE SIGN OF R.H.S.
** IT3          TEMP. LOC. STORES THE ABSOLUTE VALUE OF R.H.S.
** W            THE VALUES OF THE FORMULA WITH PUTTING Q VALUES
** WSIGN        THE SIGNS OF W ARRAY
** XSIGN        #-#, MINUS SIGN.
** YSIGN        #+#, PLUS SIGN.
** ZERO        #0#, CONSTANT ZERO
*****

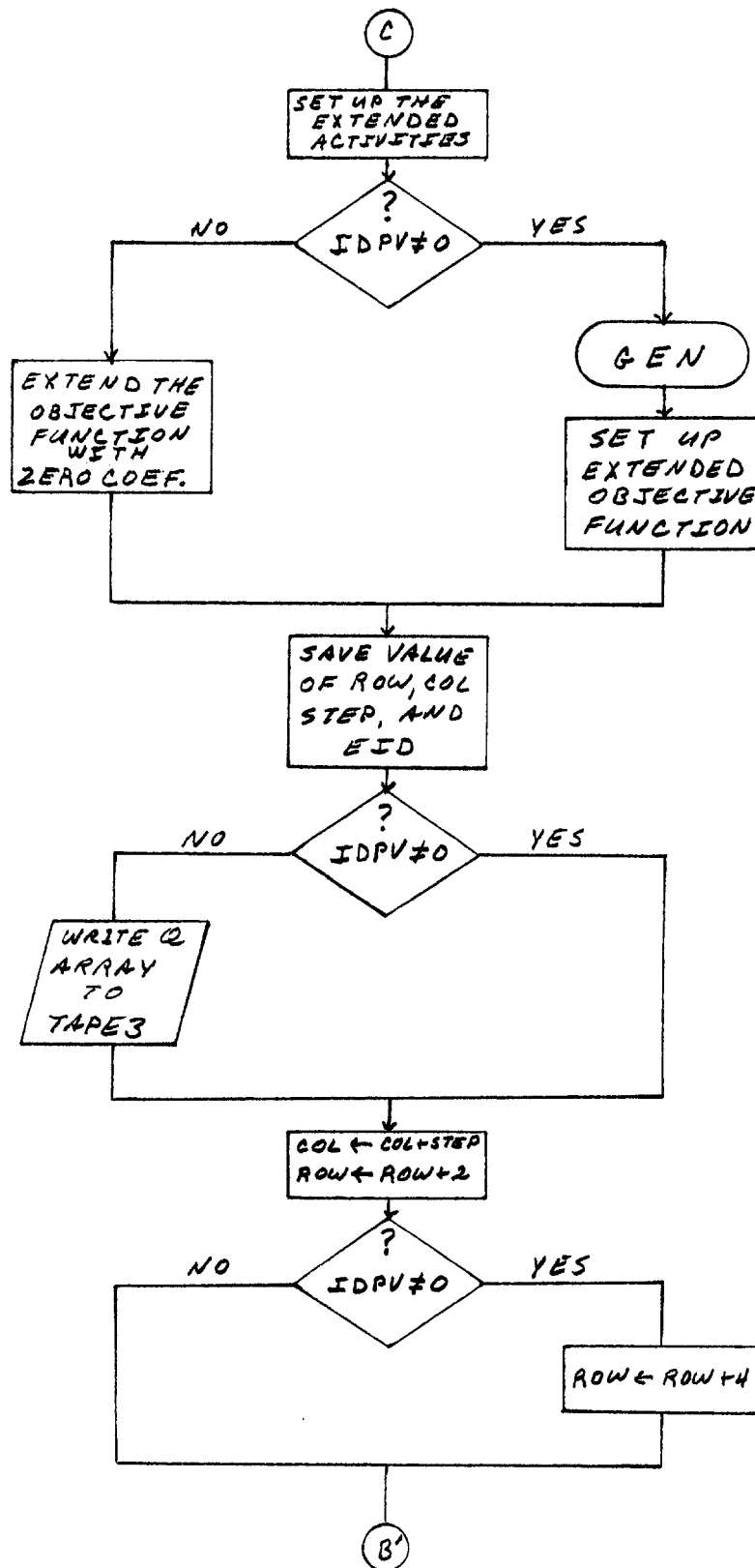
```

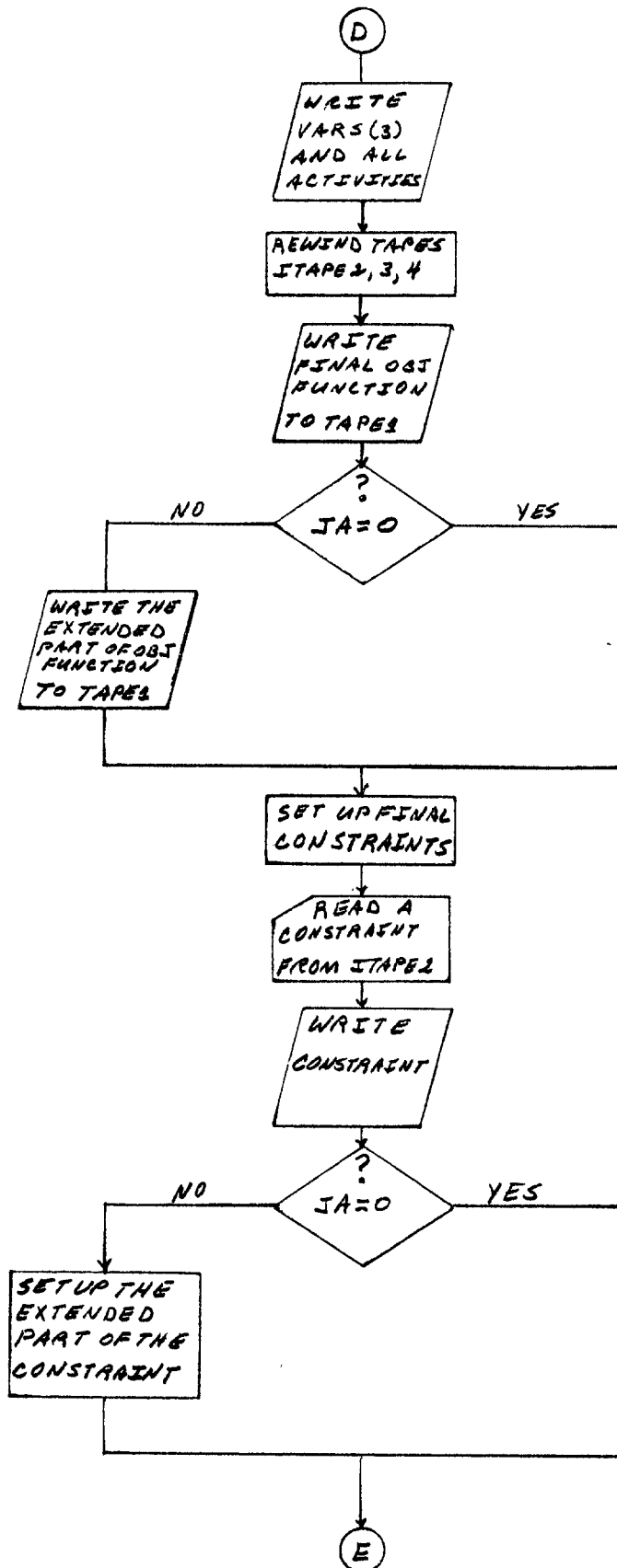
Appendix B. Flow Chart

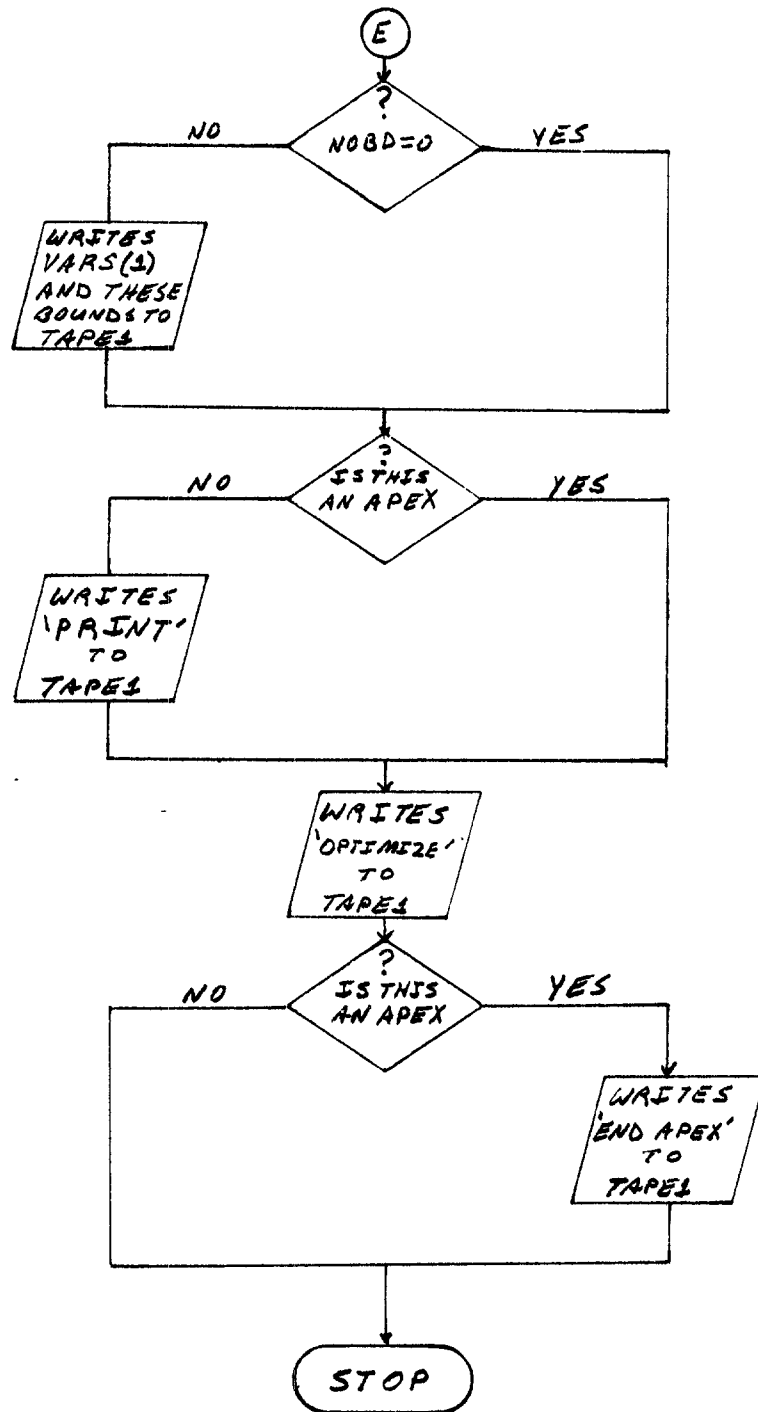


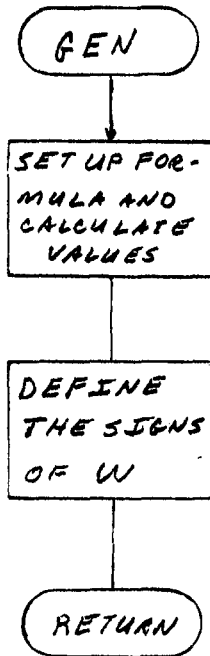


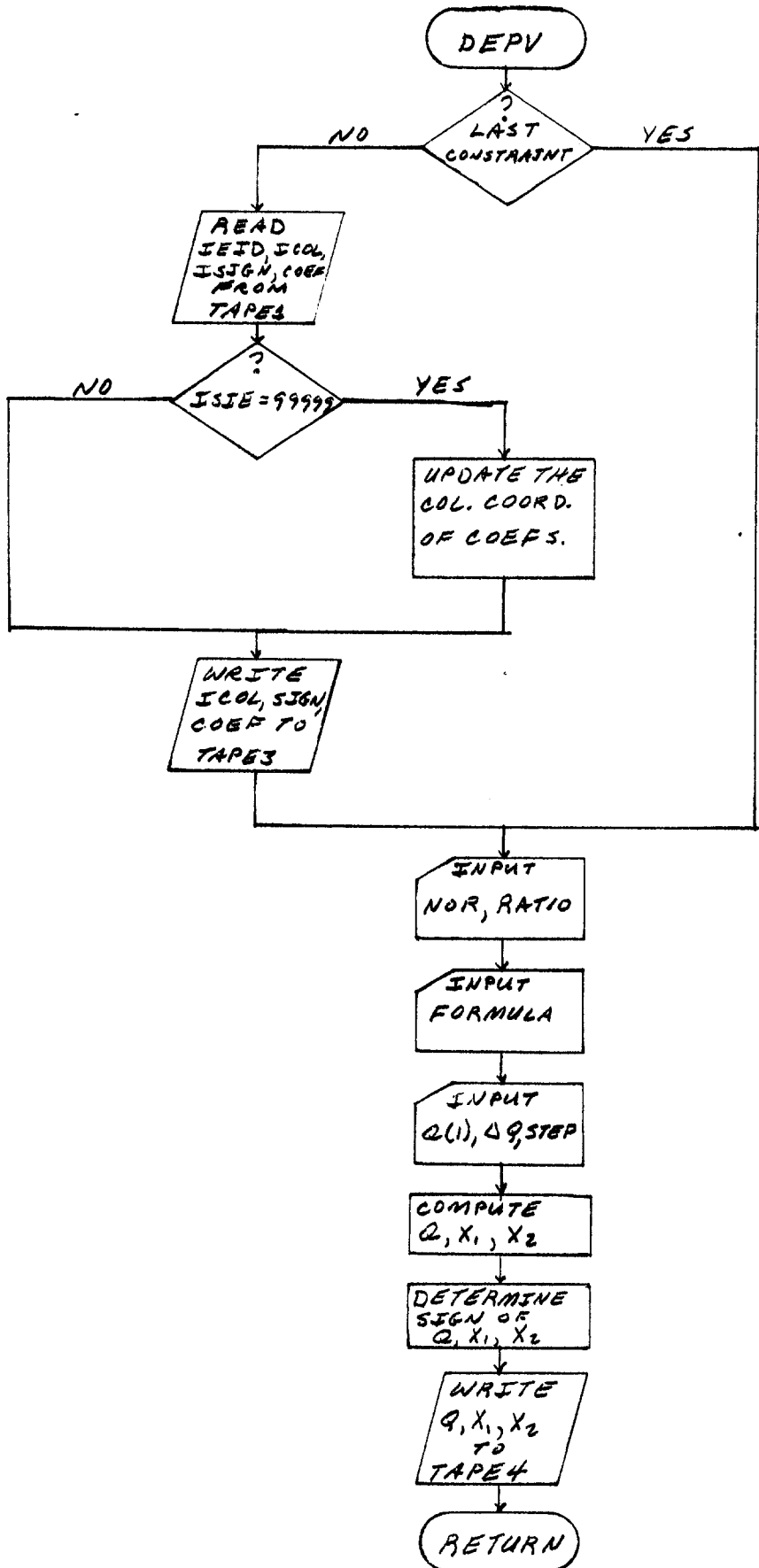


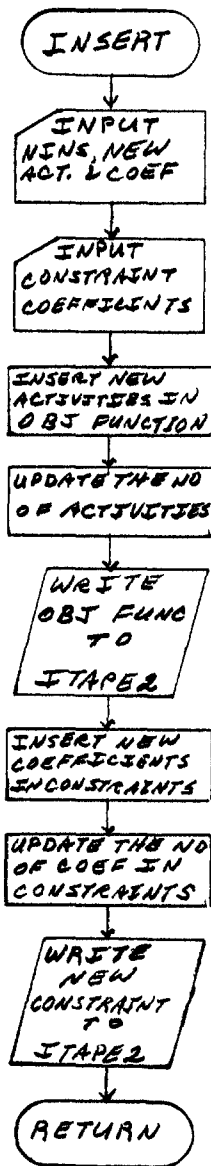












Appendix C. Program Listing

```

      MGRG      MATRIX GENERATOR AND OPTIONS
      *****
      THE FUNCTION OF THIS ROUTINE IS TO
      READ A SET OF DATA CARDS IN SPECIFIED FORMATS.
      SET UP AN M.P.O.S. MODEL OR EXACT THE MODEL BY
      CALLING AN APPROPRIATE SUBROUTINE(GEN).
      THROUGH M.P.O.S. WE CAN GET THE SOLUTIONS.
      *****
      * METHOD(N1) CONTAINS THE 12 ALGORITHMS WHICH ARE USED IN M.P.O.S.
      *
      *   N1 = 01.  #REGULAR#, 2-PHASE SIMPLEX (L.P.)
      *   N1 = 02.  #REVISED#, REVISED SIMPLEX (L.P.)
      *   N1 = 03.  #DUAL#, DUAL SIMPLEX (L.P.)
      *   N1 = 04.  #MINIT#, PRIMAL-DUAL ALG. (L.P.)
      *   N1 = 05.  #BRMIP#, BRANCH AND BOUND MIXED INT. PROGRAM. (I.P.)
      *   N1 = 06.  #DSZITP#, DIRECT SEARCH 0-1 INTEGER PROGRAM. (I.P.)
      *   N1 = 07.  #GOMORY#, GOMORY'S CUTTING PLANE (I.P.)
      *   N1 = 08.  #WOLFF#, WOLFF'S QUADRATIC SIMPLEX (Q.P.)
      *   N1 = 09.  #BEALF#, BEALF'S ALGORITHM (Q.P.)
      *   N1 = 10.  #LEMKF#, LEMKE'S COMPLEMENTARY PIVOT ALG. (Q.P.)
      *   N1 = 11.  #APEX1#, MPOS-APEX DATA FILE INTERFACE (GENFAL)
      *   N1 = 12.  #APEX2#, MPOS-APEX DATA FILE INTERFACE (GENFAL)
      *
      * VARS(N1) ARE THE RESERVED WORDS USED BY #MPOS#.
      *
      *   N1 = 1  TITLE
      *   N1 = 2  INTEGER
      *   N1 = 3  VARIABLES
      *   N1 = 4  MAXIMIZE
      *   N1 = 5  MINIMIZE
      *   N1 = 6  CONSTRAINTS
      *   N1 = 7  BOUNDS
      *   N1 = 8  PRINT
      *   N1 = 9  OPTIMIZE
      *   N1 = 10 ENDAPEX
      *   N1 = 11 BNDALI
      *   N1 = 12 BNDINT
      *   N1 = 13 RAGORI
      *   N1 = 14 RRGRRS
      *   N1 = 15 TOLERANCE
      *   N1 = 16 EPSILON
      *   N1 = 17 BNDORI
      *   N1 = 18 LIMIT
      *   N1 = 19 NOSCALE
      *   N1 = 20 CHECK
      *   N1 = 21 QCHECK
      *   N1 = 22 GO
      *   N1 = 23 STOP
      *   N1 = 24 RESCALE
      *   N1 = 25 MAXCM
      *   N1 = 26 MAXECS
      *
      * GEN(M2) CONTAIN #L.F.#, #E.Q.#, #G.E.#
      *
      *   M2 = 1.0 MEANS LESS THAN OR EQUAL TO
      *   M2 = 2.0 MEANS EQUAL TO
      *   M2 = 3.0 MEANS GREAT THAN OR EQUAL TO
  
```

```

* COWIFE AN ARRAY STORE THE COEF. OF MATRIX TO BE DUMPED
* EID EXTENDING MODEL FLAG
*      .EW 0 EXTENDING THE MODEL WITHOUT READING THE COEF.
*      (ONLY ADD ONE BLANK CARD)
*      .NE 0 EXTENDING THE MODEL WITH READING THE COEF.
*      (PUNCH THE NO. OF COEF. WHICH WILL BE ADDED.)
* ILL MAX. OR MIN. IDENTIFIER
*      = 0 MEANS MAXIMIZE
*      = 1 MEANS MINIMIZE
* IDDV 2 INDEP. VAR. FUNC. EXTENSION
*      = 0 OTHER CASE
* ISID INSERTION FLAG
*      = 0 NO NEW ACTIVITIES ARE TO BE ADDED.
*      = 99999 NEW ACTIVITIES ARE TO BE INSERTED.
* IDWIFE DUMP MATRIX TABLE FLAG
*      = 0 DONAT DUMP THE TABLE
*      = 1 DUMP THE TABLE
*      = 2 DUMP PART OF THE TABLE
* ACT ACTIVITIES
* AACT NEW ACTIVITIES
* CC COEF. OF THE FORMULA
* COFF ABSOLUTE VALUE OF THE COEFFICIENT.
* COL COLUMN NO. OF THE MATRIX(NO. OF ENTRIES)
* DELTA Q
* ICOL COLUMN COORD. OF THE ACTIVITY
* IFXP EXPONENT OF EACH ITEM OF THE FORMULA
* II NO. OF ITEMS OF THE FORMULA.
* JA NO. OF EXTENDING PROCEDURES.
* NZ NO. OF INTEGER VARIABLES FOR I.P.
* NAU NO. OF ACTIVITIES NEEDS TO BE INSERTED
* NNU TOTAL NO. OF NON-ZERO COEF.
* NINS THE PLACEMENT OF NEW ACTIVITIES WILL BE INSERTED
* NCOL ARRAY STORE THE VALUES OF COL OF EXTENDING
* NED ARRAY STORE THE VALUES OF EID OF EXTENDING
* NORD NO. OF ROUNDS
* NODD NO. OF CONSTRAINTS
* NODD ARRAY STORE THE VALUES OF NODD OF EXTENDING
* NROW ARRAY STORE THE VALUES OF ROW OF EXTENDING
* NSTEP ARRAY STORE THE VALUES OF STEP OF EXTENDING
* ONE #1, CONSTANT ONE.
* Q(1) INITIAL VALUE OF Q
* ROW ROW NO. OF THE MATRIX
* RM THE SPECIFIED R.H.S VALUE DIFFERENT FROM DEFAULT ONE
* SIGN SIGN OF THE ACTIVITY
* STEP STEPS
* TACT TEMP. LOC. STORES THE ACTIVITIES OF ROUNDS
* TCOEF TEMP. LOC. STORES THE ABSOLUTE VALUES OF BOUNDS
* TITLE TITLE OF THE PROBLEM (RESTRICTED ONE CARD)
* TSIGN TEMP. LOC. STORES THE SIGN OF BOUNDS
* TTA TEMP. LOC. STORES THE RELATIONS OF BOUNDS
* TTD TEMP. LOC. STORES THE SIGN OF R.H.S.
* TTA TEMP. LOC. STORES THE ABSOLUTE VALUE OF R.H.S.
* W THE VALUES OF THE FORMULA WITH PUTTING Q VALUES
* WSIGN THE SIGNS OF W ARRAY
* XSIGN #1, MINUS SIGN.
* YSIGN #2, PLUS SIGN.
* ZERO #0, CONSTANT ZERO

```

```

*****
PROGRAM MSAO(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4,TAPE5,TAPE6=OUT
PRINT,TAPE7,TAPE8)
REAL METHD,TEXP,L
INTEGER AICOL,AFWCOL,AEWROW
INTEGER COL,ROW,FID,STEP
COMMON /T117/Q(1000),W(1000),CC(10),TEVP(10),WSIGN(1000)
COMMON /T112/DELTA0,COL,ROW,FID,STEP,I1,IM,M1,N2,N0B0,N0B8
COMMON /T113/TITLE(8),GEL(3),ACT(1200),ICOL(1200),SIGN(1200),
* COFF(1200),METHD(10),VARS(26)
COMMON /T114/TACT(100),TT2(100),TT3(100)
COMMON /T115/NSSTEP(50),NOC0(50),NROW(50),NCOL(50),NFID(50)
COMMON /T116/Y(20,100)
COMMON /T118/RAC(10,100),C(1000),L(1000),IDPV(50)
COMMON /T119/ESIG(100),CSIGN(1000),LSIGN(1000),OSIGN(1000)
COMMON /T120/XSIGN,YSIGN,ZERO,ONE,JA,JB,JC,NRR,ITAPE1,ITAPE2
COMMON /T121/ISID,N4A,N8B,NINS,AACT(20),AICOL(20),ASIGN(20),ISIF,
* ACOFF(20),AEWROW(50),AFWSIGN(50),AEWCOFF(50),AFWCOL(50)

DATA _FRQ,FID,ISIF/0.0,-1.0/
DATA X_IGN,YSIGN,ONE,JA,JB,JC/1,1,0,0,0,1/
DATA METHD/REGULAR #,REVISED #,NDJAI #,MINIT #,
* DBMIP #,DSZ1TP #,IGOMORY #,WOLFF #,RFALE #,
* LFMKE #,APEX1 #,APEX2 #/
DATA G-I/2,LE,1,1,EQ,1,1,GE,1/
DATA VARS/1TITLE #,INTEGER #,VARIABLES #,MAXIMIZE #,
* MINIMIZE #,CONSTRAINT#,BOUNDS #,PRINT #,OPTIMIZE #,
* ENDAP-X #,ENDJAI #,ENDINT #,RNGORU #,RNGRHS #,
* TOLER,NCF #,EPSION #,RNDORU #,LIMIT #,NOSCALE #,
* CHECK #,NCHECK #,IGO #,STOP #,RFSCALE #,
* MAXCM #,MAXEYS #/
DATA IDPV,NOR0/1,1,0/
DATA AACT,TT2,TT3/200*1,100*0.0/

DATA /Y(1,MA),MA=1,100)/+YA1 #,+YA2 #,+YA3 #,+YA4 #,
* +YA5 #,
* +YA6 #,+YA7 #,+YA8 #,+YA9 #,+YA10 #,+YA11 #,
* +YA12 #,+YA13 #,+YA14 #,+YA15 #,+YA16 #,+YA17 #,
* +YA18 #,+YA19 #,+YA20 #,+YA21 #,+YA22 #,
* +YA23 #,+YA24 #,+YA25 #,+YA26 #,+YA27 #,+YA28 #,
* +YA29 #,+YA30 #,+YA31 #,+YA32 #,+YA33 #,+YA34 #,
* +YA35 #,+YA36 #,+YA37 #,+YA38 #,+YA39 #,+YA40 #,
* +YA41 #,+YA42 #,+YA43 #,+YA44 #,+YA45 #,+YA46 #,
* +YA47 #,+YA48 #,+YA49 #,+YA50 #,+YA51 #,+YA52 #,
* +YA53 #,+YA54 #,+YA55 #,+YA56 #,+YA57 #,+YA58 #,
* +YA59 #,+YA60 #,+YA61 #,+YA62 #,+YA63 #,+YA64 #,
* +YA65 #,+YA66 #,+YA67 #,+YA68 #,+YA69 #,+YA70 #,
* +YA71 #,+YA72 #,+YA73 #,+YA74 #,+YA75 #,+YA76 #,
* +YA77 #,+YA78 #,+YA79 #,+YA80 #,+YA81 #,+YA82 #,
* +YA83 #,+YA84 #,+YA85 #,+YA86 #,+YA87 #,+YA88 #,
* +YA89 #,+YA90 #,+YA91 #,+YA92 #,+YA93 #,+YA94 #,
* +YA95 #,+YA96 #,+YA97 #,+YA98 #,+YA99 #,+YA100 #/
DATA /Y(2,MA),MA=1,100)/+YB1 #,+YB2 #,+YB3 #,+YB4 #,

```

DATA	Y(3,MA),MA=1,100)/#YC1	#, #YC2	#, #YC3	#, #YC4	#,	
#YB5	#,					
#	#YB6	#, #YB7	#, #YB8	#, #YB9	#, #YB10	#, #YB11
#	#YB12	#, #YB13	#, #YB14	#, #YB15	#, #YB16	#, #YB17
#	#YB18	#, #YB19	#, #YB20	#,	#YB21	#, #YB22
#	#YB23	#, #YB24	#, #YB25	#, #YB26	#, #YB27	#, #YB28
#	#YB29	#, #YB30	#, #YB31	#, #YB32	#, #YB33	#, #YB34
#	#YB35	#, #YB36	#, #YB37	#, #YB38	#, #YB39	#, #YB40
#	#YB41	#, #YB42	#, #YB43	#, #YB44	#, #YB45	#, #YB46
#	#YB47	#, #YB48	#, #YB49	#, #YB50	#, #YB51	#, #YB52
#	#YB53	#, #YB54	#, #YB55	#, #YB56	#, #YB57	#, #YB58
#	#YB59	#, #YB60	#, #YB61	#, #YB62	#, #YB63	#, #YB64
#	#YB65	#, #YB66	#, #YB67	#, #YB68	#, #YB69	#, #YB70
#	#YB71	#, #YB72	#, #YB73	#, #YB74	#, #YB75	#, #YB76
#	#YB77	#, #YB78	#, #YB79	#, #YB80	#, #YB81	#, #YB82
#	#YB83	#, #YB84	#, #YB85	#, #YB86	#, #YB87	#, #YB88
#	#YB89	#, #YB90	#, #YB91	#, #YB92	#, #YB93	#, #YB94
#	#YB95	#, #YB96	#, #YB97	#, #YB98	#, #YB99	#, #YB100
DATA	Y(3,MA),MA=1,100)/#YC1	#, #YC2	#, #YC3	#, #YC4	#,	
#YC5	#,					
#	#YC6	#, #YC7	#, #YC8	#, #YC9	#, #YC10	#, #YC11
#	#YC12	#, #YC13	#, #YC14	#, #YC15	#, #YC16	#, #YC17
#	#YC18	#, #YC19	#, #YC20	#,	#YC21	#, #YC22
#	#YC23	#, #YC24	#, #YC25	#, #YC26	#, #YC27	#, #YC28
#	#YC29	#, #YC30	#, #YC31	#, #YC32	#, #YC33	#, #YC34
#	#YC35	#, #YC36	#, #YC37	#, #YC38	#, #YC39	#, #YC40
#	#YC41	#, #YC42	#, #YC43	#, #YC44	#, #YC45	#, #YC46
#	#YC47	#, #YC48	#, #YC49	#, #YC50	#, #YC51	#, #YC52
#	#YC53	#, #YC54	#, #YC55	#, #YC56	#, #YC57	#, #YC58
#	#YC59	#, #YC60	#, #YC61	#, #YC62	#, #YC63	#, #YC64
#	#YC65	#, #YC66	#, #YC67	#, #YC68	#, #YC69	#, #YC70
#	#YC71	#, #YC72	#, #YC73	#, #YC74	#, #YC75	#, #YC76
#	#YC77	#, #YC78	#, #YC79	#, #YC80	#, #YC81	#, #YC82
#	#YC83	#, #YC84	#, #YC85	#, #YC86	#, #YC87	#, #YC88
#	#YC89	#, #YC90	#, #YC91	#, #YC92	#, #YC93	#, #YC94
#	#YC95	#, #YC96	#, #YC97	#, #YC98	#, #YC99	#, #YC100
DATA	Y(4,MA),MA=1,100)/#YD1	#, #YD2	#, #YD3	#, #YD4	#,	
#YD5	#,					
#	#YD6	#, #YD7	#, #YD8	#, #YD9	#, #YD10	#, #YD11
#	#YD12	#, #YD13	#, #YD14	#, #YD15	#, #YD16	#, #YD17
#	#YD18	#, #YD19	#, #YD20	#,	#YD21	#, #YD22
#	#YD23	#, #YD24	#, #YD25	#, #YD26	#, #YD27	#, #YD28
#	#YD29	#, #YD30	#, #YD31	#, #YD32	#, #YD33	#, #YD34
#	#YD35	#, #YD36	#, #YD37	#, #YD38	#, #YD39	#, #YD40
#	#YD41	#, #YD42	#, #YD43	#, #YD44	#, #YD45	#, #YD46
#	#YD47	#, #YD48	#, #YD49	#, #YD50	#, #YD51	#, #YD52
#	#YD53	#, #YD54	#, #YD55	#, #YD56	#, #YD57	#, #YD58
#	#YD59	#, #YD60	#, #YD61	#, #YD62	#, #YD63	#, #YD64
#	#YD65	#, #YD66	#, #YD67	#, #YD68	#, #YD69	#, #YD70
#	#YD71	#, #YD72	#, #YD73	#, #YD74	#, #YD75	#, #YD76
#	#YD77	#, #YD78	#, #YD79	#, #YD80	#, #YD81	#, #YD82
#	#YD83	#, #YD84	#, #YD85	#, #YD86	#, #YD87	#, #YD88
#	#YD89	#, #YD90	#, #YD91	#, #YD92	#, #YD93	#, #YD94
#	#YD95	#, #YD96	#, #YD97	#, #YD98	#, #YD99	#, #YD100
DATA	Y(5,MA),MA=1,100)/#YE1	#, #YE2	#, #YE3	#, #YE4	#,	
#YF5	#,					

*	YF6	Z,YF7	Z,YF8	Z,YF9	Z,YF10	Z,YF11	Z.
*	YF12	Z,YF13	Z,YF14	Z,YF15	Z,YF16	Z,YF17	Z.
*	YF18	Z,YF19	Z,YF20	Z.	YF21	Z,YF22	Z.
*	YF23	Z,YF24	Z,YF25	Z,YF26	Z,YF27	Z,YF28	Z.
*	YF29	Z,YF30	Z,YF31	Z,YF32	Z,YF33	Z,YF34	Z.
*	YF35	Z,YF36	Z,YF37	Z,YF38	Z,YF39	Z,YF40	Z.
*	YF41	Z,YF42	Z,YF43	Z,YF44	Z,YF45	Z,YF46	Z.
*	YF47	Z,YF48	Z,YF49	Z,YF50	Z,YF51	Z,YF52	Z.
*	YF53	Z,YF54	Z,YF55	Z,YF56	Z,YF57	Z,YF58	Z.
*	YF59	Z,YF60	Z,YF61	Z,YF62	Z,YF63	Z,YF64	Z.
*	YF65	Z,YF66	Z,YF67	Z,YF68	Z,YF69	Z,YF70	Z.
*	YF71	Z,YF72	Z,YF73	Z,YF74	Z,YF75	Z,YF76	Z.
*	YF77	Z,YF78	Z,YF79	Z,YF80	Z,YF81	Z,YF82	Z.
*	YF83	Z,YF84	Z,YF85	Z,YF86	Z,YF87	Z,YF88	Z.
*	YF89	Z,YF90	Z,YF91	Z,YF92	Z,YF93	Z,YF94	Z.
*	YF95	Z,YF96	Z,YF97	Z,YF98	Z,YF99	Z,YF100	Z/
DATA	Y(6,MA).MA=1,100)/YF1	Z,YF2	Z,YF3	Z,YF4	Z,YF5	Z.	
Z,YF5	Z.						
*	YF6	Z,YF7	Z,YF8	Z,YF9	Z,YF10	Z,YF11	Z.
*	YF12	Z,YF13	Z,YF14	Z,YF15	Z,YF16	Z,YF17	Z.
*	YF18	Z,YF19	Z,YF20	Z.	YF21	Z,YF22	Z.
*	YF23	Z,YF24	Z,YF25	Z,YF26	Z,YF27	Z,YF28	Z.
*	YF29	Z,YF30	Z,YF31	Z,YF32	Z,YF33	Z,YF34	Z.
*	YF35	Z,YF36	Z,YF37	Z,YF38	Z,YF39	Z,YF40	Z.
*	YF41	Z,YF42	Z,YF43	Z,YF44	Z,YF45	Z,YF46	Z.
*	YF47	Z,YF48	Z,YF49	Z,YF50	Z,YF51	Z,YF52	Z.
*	YF53	Z,YF54	Z,YF55	Z,YF56	Z,YF57	Z,YF58	Z.
*	YF59	Z,YF60	Z,YF61	Z,YF62	Z,YF63	Z,YF64	Z.
*	YF65	Z,YF66	Z,YF67	Z,YF68	Z,YF69	Z,YF70	Z.
*	YF71	Z,YF72	Z,YF73	Z,YF74	Z,YF75	Z,YF76	Z.
*	YF77	Z,YF78	Z,YF79	Z,YF80	Z,YF81	Z,YF82	Z.
*	YF83	Z,YF84	Z,YF85	Z,YF86	Z,YF87	Z,YF88	Z.
*	YF89	Z,YF90	Z,YF91	Z,YF92	Z,YF93	Z,YF94	Z.
*	YF95	Z,YF96	Z,YF97	Z,YF98	Z,YF99	Z,YF100	Z/
DATA	Y(7,MA).MA=1,100)/YG1	Z,YG2	Z,YG3	Z,YG4	Z,YG5	Z.	
Z,YG5	Z.						
*	YG6	Z,YG7	Z,YG8	Z,YG9	Z,YG10	Z,YG11	Z.
*	YG12	Z,YG13	Z,YG14	Z,YG15	Z,YG16	Z,YG17	Z.
*	YG18	Z,YG19	Z,YG20	Z.	YG21	Z,YG22	Z.
*	YG23	Z,YG24	Z,YG25	Z,YG26	Z,YG27	Z,YG28	Z.
*	YG29	Z,YG30	Z,YG31	Z,YG32	Z,YG33	Z,YG34	Z.
*	YG35	Z,YG36	Z,YG37	Z,YG38	Z,YG39	Z,YG40	Z.
*	YG41	Z,YG42	Z,YG43	Z,YG44	Z,YG45	Z,YG46	Z.
*	YG47	Z,YG48	Z,YG49	Z,YG50	Z,YG51	Z,YG52	Z.
*	YG53	Z,YG54	Z,YG55	Z,YG56	Z,YG57	Z,YG58	Z.
*	YG59	Z,YG60	Z,YG61	Z,YG62	Z,YG63	Z,YG64	Z.
*	YG65	Z,YG66	Z,YG67	Z,YG68	Z,YG69	Z,YG70	Z.
*	YG71	Z,YG72	Z,YG73	Z,YG74	Z,YG75	Z,YG76	Z.
*	YG77	Z,YG78	Z,YG79	Z,YG80	Z,YG81	Z,YG82	Z.
*	YG83	Z,YG84	Z,YG85	Z,YG86	Z,YG87	Z,YG88	Z.
*	YG89	Z,YG90	Z,YG91	Z,YG92	Z,YG93	Z,YG94	Z.
*	YG95	Z,YG96	Z,YG97	Z,YG98	Z,YG99	Z,YG100	Z/
DATA	Y(8,MA).MA=1,100)/YH1	Z,YH2	Z,YH3	Z,YH4	Z,YH5	Z.	

DATA	Y(9,MA),MA=1,100)/#YI1	#YI2	#YI3	#YI4	#YI5	#YI6	#YI7
#YI5	#	#	#	#	#	#	#
*	YH0	#YH7	#YH8	#YH9	#YH10	#YH11	#
*	YH12	#YH13	#YH14	#YH15	#YH16	#YH17	#
*	YH18	#YH19	#YH20	#	#YH21	#YH22	#
*	YH23	#YH24	#YH25	#YH26	#YH27	#YH28	#
*	YH29	#YH30	#YH31	#YH32	#YH33	#YH34	#
*	YH35	#YH36	#YH37	#YH38	#YH39	#YH40	#
*	YH41	#YH42	#YH43	#YH44	#YH45	#YH46	#
*	YH47	#YH48	#YH49	#YH50	#YH51	#YH52	#
*	YH53	#YH54	#YH55	#YH56	#YH57	#YH58	#
*	YH59	#YH60	#YH61	#YH62	#YH63	#YH64	#
*	YH65	#YH66	#YH67	#YH68	#YH69	#YH70	#
*	YH71	#YH72	#YH73	#YH74	#YH75	#YH76	#
*	YH77	#YH78	#YH79	#YH80	#YH81	#YH82	#
*	YH83	#YH84	#YH85	#YH86	#YH87	#YH88	#
*	YH89	#YH90	#YH91	#YH92	#YH93	#YH94	#
*	YH95	#YH96	#YH97	#YH98	#YH99	#YH100	#
DATA	Y(10,MA),MA=1,100)/#YJ1	#YJ2	#YJ3	#YJ4	#YJ5	#YJ6	#YJ7
#YJ5	#	#	#	#	#	#	#
*	YJ0	#YJ7	#YJ8	#YJ9	#YJ10	#YJ11	#
*	YJ12	#YJ13	#YJ14	#YJ15	#YJ16	#YJ17	#
*	YJ18	#YJ19	#YJ20	#	#YJ21	#YJ22	#
*	YJ23	#YJ24	#YJ25	#YJ26	#YJ27	#YJ28	#
*	YJ29	#YJ30	#YJ31	#YJ32	#YJ33	#YJ34	#
*	YJ35	#YJ36	#YJ37	#YJ38	#YJ39	#YJ40	#
*	YJ41	#YJ42	#YJ43	#YJ44	#YJ45	#YJ46	#
*	YJ47	#YJ48	#YJ49	#YJ50	#YJ51	#YJ52	#
*	YJ53	#YJ54	#YJ55	#YJ56	#YJ57	#YJ58	#
*	YJ59	#YJ60	#YJ61	#YJ62	#YJ63	#YJ64	#
*	YJ65	#YJ66	#YJ67	#YJ68	#YJ69	#YJ70	#
*	YJ71	#YJ72	#YJ73	#YJ74	#YJ75	#YJ76	#
*	YJ77	#YJ78	#YJ79	#YJ80	#YJ81	#YJ82	#
*	YJ83	#YJ84	#YJ85	#YJ86	#YJ87	#YJ88	#
*	YJ89	#YJ90	#YJ91	#YJ92	#YJ93	#YJ94	#
*	YJ95	#YJ96	#YJ97	#YJ98	#YJ99	#YJ100	#
DATA	Y(10,MA),MA=1,100)/#YJ1	#YJ2	#YJ3	#YJ4	#YJ5	#YJ6	#YJ7
#YJ5	#	#	#	#	#	#	#
*	YJ0	#YJ7	#YJ8	#YJ9	#YJ10	#YJ11	#
*	YJ12	#YJ13	#YJ14	#YJ15	#YJ16	#YJ17	#
*	YJ18	#YJ19	#YJ20	#	#YJ21	#YJ22	#
*	YJ23	#YJ24	#YJ25	#YJ26	#YJ27	#YJ28	#
*	YJ29	#YJ30	#YJ31	#YJ32	#YJ33	#YJ34	#
*	YJ35	#YJ36	#YJ37	#YJ38	#YJ39	#YJ40	#
*	YJ41	#YJ42	#YJ43	#YJ44	#YJ45	#YJ46	#
*	YJ47	#YJ48	#YJ49	#YJ50	#YJ51	#YJ52	#
*	YJ53	#YJ54	#YJ55	#YJ56	#YJ57	#YJ58	#
*	YJ59	#YJ60	#YJ61	#YJ62	#YJ63	#YJ64	#
*	YJ65	#YJ66	#YJ67	#YJ68	#YJ69	#YJ70	#
*	YJ71	#YJ72	#YJ73	#YJ74	#YJ75	#YJ76	#
*	YJ77	#YJ78	#YJ79	#YJ80	#YJ81	#YJ82	#
*	YJ83	#YJ84	#YJ85	#YJ86	#YJ87	#YJ88	#
*	YJ89	#YJ90	#YJ91	#YJ92	#YJ93	#YJ94	#
*	YJ95	#YJ96	#YJ97	#YJ98	#YJ99	#YJ100	#

```

*** INPUT THE FLAG, NO. OF COLUMNS AND ROWS.
    READ (2,500) ID,M,M1,COL,ROW

*** INPUT THE TITLE OF YOUR PROBLEM.
    READ (2,501) TITLE

*** WRITE OUT THE ALGORITHM'S NAME AND TITLE
    WRITE (1,100) METHOD(M1),VARS(1),TITLE

*** TEST WHETHER THIS IS AN I.P. PROBLEM.
    IF IT IS AN I.P. PROBLEM, READ THE INT. VARS AND WRITE TO TAPE 1.
    O.W. CONTINUE THE PROCESS.
    IF ((M1 .LT. 5) .OR. (M1 .GT. 7)) GO TO 6
    READ (2,503) N2,(ACT(I1),I1=1,N2)
    WRITE (1,101) VARS(2),(ACT(I1),I1=1,N2)
    5 CONTINUE

*** READ I. TO ALL OF THE ACTIVITIES
    READ (2,505) (ACT(IA),IA=1,COL)

*** READ INTO THE COEFFICIENTS OF OBJECT FUNCTION.
    J1=1
10 CONTINUE
    READ (2,506) (ICOL (IB),SIGN(IB),COFF (IB),IB=J1,J1+4)
    DO 15 JC=J1,J1+4
    IF (ICOL (IC) .EQ. -999) GO TO 20
15 CONTINUE
    J1=J1+4
    GO TO 10

*** NOOB IS THE NO. OF NON-ZERO COEFF. OF OBJECTIVE FUNCTION.
20 CONTINUE
    NOOB = IC - 1

*** WRITE OUT THE COEFF. OF OBJ. FUNCTION TO PERMANENT FILE (TAPE 2)
    WRITE (2,201) (SIGN(ID),COFF (ID),ICOL (ID),ID=1,NOOB)

```

```

*** TEST IF THERE IS NO ANY CONSTRAINT IN NON-ARGUMENTED
CONSTRAINT MATRIX.
IF (ROWL.FN. 0) GO TO 43

*** READ 1. TO THE COEFFIENTS OF CONSTRAINTS.
J2=1 & K1=0 & IEW=0
26 CONTINUE
DO 30 J2=1,ROW
27 CONTINUE
READ (-,506) (ICOL(IF),SIGN(IF),COEF(IF),IF=J2,J2+4)
DO 30 J2=J2+4
IF (ICOL(IG) .LT. 0) GO TO 40
30 CONTINUE
J2=J2+4
GO TO 27
40 CONTINUE

*** TEST IF THIS ROW CONTAINS ONLY ONE COEF. WITH  $\neq 0$  RELATION.
IF YES, THEN PUT IT IN TEMP. LOC.
OTHERW. CONTINUE THE PROCESS.
IF (IG.GT. 2) GO TO 45
IF (ICOL(IG) .LT. -100) GO TO 45
K1=K1+1
TACT(K1) = ACT(ICOL(1))
TT2(K1)=SIGN(IG)
TT3(K1)=COEF(IG)/COEF(1)
GO TO 48
45 CONTINUE

*** NOCO CONTAINS THE NO. OF NON-ZERO COEF. OF CONSTRAINT.
IEW=IEW+1
NOCO(1-IEW) = IG - 1

*** WRITE OUT THE COEF. AND R.H.S. OF EACH CONSTRAINT.
WRITE (2,201) (SIGN(IH),COEF(IH),ICOL(IH),IH=1,NOCO(IEW))
IGFL=ICOL(IG)/(-100)
WRITE (2,202) GEL(IGFL),SIGN(IG),COEF(IG)
48 CONTINUE
J2=1
50 CONTINUE

*** NORD CONTAINS THE NO. OF BOUNDS.
*** NOCO CONTAINS THE NO. OF CONSTRAINTS.
NORD = K1 & NOCO = ROW-NORD

```

```

43 CONTINUE
*** TEST IF THERE ARE SOME NEW ACTIVITIES NEED TO BE INSERTED INTO
    THE FOUND MATRIX. (ISID = 99999)
    IF YES, THEN CALL SUBROUTINE #INSERT#
    OTHERWISE, CONTINUE THE NORMAL PROCESS.
    LOOP = 1
    ITAPE2 = 2
44 CONTINUE
    READ (2,511) ISID
    IF (EQ(5) .NE. 0.) GO TO 45
    IF (ISID .NE. 99999) GO TO 52
    ILOOP = LOOP/2  ILOOP=ILOOP*2  SITAPE1 = 2  SITAPE2 = 7
    IF (ILOOP .NE. LOOP) GO TO 44
    ITAPE1 = 7  SITAPE2 = 2
45 CONTINUE
    CALL #INSERT#
    GO TO 44
52 CONTINUE

*** TEST IF THE PROBLEM NEEDS TO BE EXTENDED (CHECK EOF)
53 CONTINUE
    JAC=JA+.
    READ (2,510) EID,RM ,IOPV(JA)
    IF (EQ(5) .NE. 0.) GO TO 45
    IF (ABS(RM) .LE. 10.**(-10)) GO TO 54
    ONE = RM
54 CONTINUE

*** TEST IF THIS IS AN EXTENSION OF TWO DEP. VAR. FUNCTIONS
    IF YES, THEN CALL SUBROUTINE #DEPVA#
    O.W. CONTINUE THE PROCESS.
    WRITE (3,507) FID,ONE
    IF (IDAV(JA) .EQ. 0) GO TO 55
    CALL #DEPVA#
    GO TO 47
55 CONTINUE
    IF (EID .EQ. 0) GO TO 56
    READ (2,508) (ICOL(IT),SIGN(IT),COFF(IT),IT=1,ETD)

*** UPDATE THE COL. COORD. OF NON-ZERO COEF. IF COL. .GT. NINS.
    IF (IS.F .NE. 99999) GO TO 46
    DO 42 I=1,FID
    IF (ICOL(IT) .GT. NINS) ICOL(IT)=ICOL(IT)+NAA
42 CONTINUE
46 CONTINUE
    WRITE (3,201) (SIGN(IT),COFF(IT),ICOL(IT),IT=1,FID)
56 CONTINUE

```

```

*** PROCESS THE EXTENDING PROCEDURE.
READ INTO THE FORMULA
READ (2,500) IO,(LC(IK),IEVP(IK),IK=1,IO)

*** READ THE INITIAL VALUE OF  $\alpha$ , DELTA  $\alpha$  AND STEPS.
READ (2,500) Q(1),DELTAQ,STEP

*** SET UP THE NEW ACTIVITIES.
57 CONTINUE
DO 58 J=1,STEP
  ACT(KA,COL)=Y(JA,KK)
58 CONTINUE

*** CALL THE SUBROUTINE GEN TO GENERATE THE VALUES OF  $\alpha$  AND  $w$ .
IF (IDNV(JA) .NE. 0) GO TO 62
CALL GEN

*** EXTENDING THE OBJECT FUNCTION.
  IQ=1
  DO 60 JP=COL+1,COL+STEP
    SIGN(IQ)=SIGN(IQ)
    COEF(IQ)=ABS(W(IQ)) $ ICOL(IP)=COL+IQ
    IQ=IQ+1
  60 CONTINUE
  GO TO 64

  62 CONTINUE
  IQ=1
  DO 63 JP=COL+1,COL+STEP
    SIGN(IQ) = 1+1
    COEF(IQ) = 0.0 $ ICOL(IP) = COL+IQ
    IQ = IQ+1
  63 CONTINUE

*** STORE THE VALUES OF ROW ,COL, AND STEPS.
  64 CONTINUE
  NROW(JA)=ROW $ NCOL(JA)=COL $ NSTEP(JA)=STEP $ NFIn(JA)=EIN

*** EXTENDING THE CONSTRAINTS.
IF (IDNV(JA) .NE. 0) GO TO 66
WRITE (3,500) (W(IT),IT=1,NSTEP(JA))

```

*** UPDATE THE VALUES OF COL AND ROW.

```
65 CONTINUE
   COL=COL+STEP      ROW = ROW+2
   IF (IDM(JA) .NE. 0) ROW = ROW+4
   GO TO 63
```

```
65 CONTINUE
   JAJ=JA-
```

*** SET UP THE VARIABLES.

```
   WRITE (1,103) VARS(3)
   WRITE (1,504) (ACT(JN),JN=1,COL)
```

*** REWIND THE 3 PERMANENT FILES (ITAPE2,TAPE3,TAPE4)

```
   REWIND ITAPE2
   REWIND 3
   REWIND 4
```

*** SET UP THE FINAL OBJECT FUNCTION.

```
   IDM=IDM+4
   WRITE (1,103) VARS(IDM)
   READ (7,TAPE2,201) (SIGN(ID),COFF(ID),ICOL(ID),ID=1,N00B)
   WRITE (1,102) (SIGN(ID),COFF(ID),ACT(ICOL(ID)),ID=1,N00B)
   IF (JA .EQ. 0) GO TO 66
   WRITE (1,102) (SIGN(ID),COFF(ID),ACT(ICOL(ID)),ID=NCOL(1)+1,COL)
```

*** SET UP THE FINAL CONSTRAINTS.

```
66 CONTINUE
   WRITE (1,103) VARS(6)
   DO 70 IC=1,N00C
     READ (7,TAPE2,201) (SIGN(IH),COFF(IH),ICOL(IH),IH=1,N00C(JC))
     WRITE (1,102) (SIGN(IH),COFF(IH),ACT(ICOL(IH)),IH=1,N00C(JC))
     READ (7,TAPE2,202) TTT1,TTT2,TTT3
     WRITE (1,202) TTT1,TTT2,TTT3
70 CONTINUE
   IF (JA .EQ. 0) GO TO 85
   IAW = .COL(1)
   DO 80 JK=1,JA
     READ (7,507) FID,ONF
     IF (IDM(JK) .EQ. 0) GO TO 74
     DO 840 I3=1,FID
       READ (7,511) IFID(I3)
       IIF = .FID(I3) + 1
       READ (7,508) (ICOL(I1),SIGN(I1),COFF(I1),I1=1,IIF)
       WRITE (1,102) (SIGN(I1),COFF(I1),ACT(ICOL(I1)),I1=1,IFID(I3))
```

```

ICT = .COL(IIE)/(-100)
READ (7,540) (CSIGN(IH),C(IH),IH=1,NSTEP(JK))
WRITE (1,102) (CSIGN(IH),C(IH),ACT(NCOL(JK)+IH),IH=1,NSTEP(JK))
WRITE (1,202) GEL(ICT),SIGN(IIF),COEF(IIE)
840 CONTINUE
DONE = 1.0
WRITE (1,102) (YSIGN,DONE,ACT(NCOL(JK)+I2),I2=1,NSTEP(JK))
WRITE (1,202) GEL(1),YSIGN,DONE
GO TO 40
74 CONTINUE
IF (ELC.EQ. 0) GO TO 75
READ (7,201) (SIGN(II),COEF(II),ICOL(II),II=1,ETD)
WRITE (1,102) (SIGN(II),COEF(II),ACT(ICOL(II)),II=1,ETD)
75 CONTINUE
READ (7,300) (N(II),II=1,NSTEP(JK))
WRITE (1,102) (XSIGN,Q(II-NCOL(JK)),ACT(II),II=NCOL(JK)+1,NCOL(JK)+
.NSTEP(JK))
I=3
IWW=NS-EP(JK)/2 + 1 + IWW
IF (SIGN(IWW).EQ. 7+7) GO TO 77
I=1
77 CONTINUE
IWW = IWW+NSTEP(JK)/2-1
WRITE (1,202) GEL(IV),YSIGN,ZERO
WRITE (1,102) (YSIGN,DONE,ACT(II),II=NCOL(JK)+1,NCOL(JK)+NSTEP(JK))
WRITE (1,202) GEL(1),YSIGN,DONE
80 CONTINUE

*** IF NOBA EQUAL TO 0, THEN THERE IS NO BOUND EXITS.
ELSE PRINT OUT THE VERB #BOUND# AND SET UP THE BOUND SECTION.
85 CONTINUE
IF (NOBD.EQ. 0) GO TO 90
WRITE (1,103) VARS(7)
DO 88 KR=1,NOBD
WRITE (1,104) TACT(KR),GEL(1),TT2(KR),TT3(KR)
88 CONTINUE
90 CONTINUE

*** PRINT OUT THE VERB #OPTIMIZE#.
WRITE (1,103) VARS(9)

*** IF IT IS FOR MPDS-APEX, THEN PRINT #ENDAPEX# AND STOP
ELSE STOP.
IF ((M.EQ. 11) .AND. (M1.EQ. 12)) STOP
WRITE (1,103) VARS(10)

*** FORMAT.

```

```

00 FORMAT (A10/A10/A410)
01 FORMAT (A10/A(A7,3))
02 FORMAT (3(A1,F16.5,A7))
03 FORMAT (A10)
04 FORMAT (A7,A10,A1,F16.5)
01 FORMAT (5(A1,F16.5,I4))
02 FORMAT (A10,A1,F16.5)
00 FORMAT (8F16.2)
00 FORMAT (11,I2,2I5)
01 FORMAT (A410)
02 FORMAT (F16.2)
03 FORMAT (13,11A7/(3X,11A7))
04 FORMAT (A(3X,A7))
05 FORMAT (3X,11A7)
06 FORMAT (5(I4,A1,F11.2))
07 FORMAT (15,F16.5)
08 FORMAT (15,5(F10.4,F5.0))
09 FORMAT (2F10.4,I5)
10 FORMAT (15,F10.2,I6)
11 FORMAT (I5)
99 FORMAT (5(A1,F15.3))
STOP
END

```

```

SUBROUTINE GEN
INTEGER COL,ROW,FTD,STEP
REAL W,TIM,TEXP,C
COMMON /T111/Q(1000),W(1000),CC(10),TEXP(10),SIGN(1000)
COMMON /T112/DELTA9,COL,ROW,FTD,STEP,I1,1DM,M1,N2,N080,N008

```

```

*** SET UP AND CALCULATE THE FORMULA
DO 8 I=1,STEP
  W(I)=A.0
  DO 3 J=1,10
    W(I) = W(I) + CC(J)*N(I)**TEXP(J)
  * CONTINUE
  Q(I+1,Q(I))+DELTA9
  * CONTINUE

```

```

*** DEFINE THE SIGNS OF W.
DO 13 N=1,STEP
  *SIGN(N)=A.0
  IF (A(N).GE.0.0) GO TO 13
  *SIGN(N)=A.-4
13 CONTINUE
RETURN
END

```



```

SUBROUTINE DEPV
  INTEGER COL,ROW,FI0,STEP
  REAL METHD,TEXP,L
  COMMON /T111/Q(1000),A(1000),C0(10),IEXP(10),JSIGN(1000)
  COMMON /T112/DELTA0,COL,ROW,FI0,STEP,I1,I04,M1,M2,N0BD,N0nR
  COMMON /T113/TIME(8),GEL(3),ACT(1200),ICOL(1200),SIGN(1200),
  * COFF(1200),METHD(12),VARS(26)
  COMMON /T118/RATIO(100),C(1000),L(1000),IDPV(50)
  COMMON /T119/IEIN(100),CSIGN(1000),LSTGN(1000),ACTGN(1000)
  COMMON /T121/ISIF,NAA,NB0,NINS,AACT(20),ALCOL(20),ASIGN(20),ISIF,
  * ACOFF(20),AEWROW(50),AFWSIGN(50),AENCOFF(50),AFWCOL(50)

  *** READ I. TO THE NO. OF NON-ZERO COEF. OF EACH ROW
  DO 800 I3=1,FI0
    READ (2,507) IFID(I3)
    READ (2,506) (ICOL(I1),SIGN(I1),COFF(I1),I1=1,IEIN(I3)+1)

  *** UPDATE THE COL. COORD. OF NON-ZERO COEF. IF COL. > MINS.
  IF (ISIF.NE. 99999) GO TO 701
  DO 700 I1=1,IEIN(I3)
    IF (ICOL(I1) > MINS) ICOL(I1)=ICOL(I1)+NAA
  700 CONTINUE
  701 CONTINUE

  *** WRITE OUT TO PERMANENT FILE (TAPE 3)
  WRITE (3,507) IFID(I3)
  WRITE (3,506) (ICOL(I1),SIGN(I1),COFF(I1),I1=1,IEIN(I3)+1)
  800 CONTINUE

  *** READ I. TO THE NO. OF RATIOS DESIRED (NGR) AND RATIOS (RATIO)
  READ (2,504) NGR,(RATIO(I2),I2=1,NGR)

  *** READ I. TO THE FORMULA.
  READ (2,505) AA,ALPH1,ALPH2
  XXX = ALPH1 + ALPH2
  XAX = 1/XXX
  XXH = LALPH1/XXX

  *** READ I. TO THE INITIAL VALUES OF Q, DELTA 0, AND STEPS.
  READ (2,509) Q(1),DELTA0,STEP

```

```

*** COMPUTE THE VALUES OF Q, L, AND C.
DO 800 I5=1,NOR
DO 805 I6=1,STEP
I11 = I5+STEP*I6-STEP
Q(I11+.1) = Q(I11)+DEL1AQ
L(I11) = (Q(I11)+XXA) * (AA * (-XXA)) * (RATIO(I5)*XXB)
C(I11) = RATIO(I5)*L(I11)
805 CONTINUE
Q(I11+.1) = Q(1)
800 CONTINUE

```

```

*** DETERMINE THE SIGNS OF ARRAYS Q, C, AND L.
NRR = NOR*STEP
DO 820 I1=1,NRR
QSIGN(I1) = 1
CSIGN(I1) = 1
LSIGN(I1) = 1
IF (Q(I1) .GE. 0.) GO TO 816
Q(I1) = -1
816 CONTINUE
IF (C(I1) .GE. 0.) GO TO 817
C(I1) = -1
817 CONTINUE
IF (L(I1) .GE. 0.) GO TO 818
L(I1) = -1
818 CONTINUE
820 CONTINUE
STEP = NRR

```

```

*** WRITE OUT TO PERMANENT FILE (TAPE 4)
WRITE (4,500) (CSIGN(I8),C(I8),I8=1,NRR)
WRITE (4,500) (LSIGN(I9),L(I9),I9=1,NRR)
WRITE (4,500) (QSIGN(I1),Q(I1),I1=1,NRR)

```

*** FORMATE.

```

02  FORMAT (3(41,F16.5,A7))
03  FORMAT (A10)
01  FORMAT (5(41,F16.5,I4))
02  FORMAT (A10,A1,F16.5)
05  FORMAT (3X,11A7)
06  FORMAT (5(I4,A1,F11.2))
07  FORMAT (I5)
09  FORMAT (2F10.4,I5)
84  FORMAT (I3,11F7.2/(3X,11F7.2))
85  FORMAT (3F10.2)
99  FORMAT (5(41,F15.3))
    RETURN
    END

```

SUBROUTINE INSERT

```

INTEGER COL,ROW,FID,STEP
INTEGER ATCOL,AFWCOL,AEWROW
COMMON /T112/DELTA,COL,ROW,EIN,STEP,I,IIN,M1,M2,N0BD,N0BR
COMMON /T113/TITLE(8),GEL(3),ACT(1200),ICOL(1200),SIGN(1200),
* COEF(1200),METHD(15),VARS(26)
COMMON /T115/NSTEP(50),NOC0(50),NROW(50),NCOL(50),NETD(50)
COMMON /T120/ASIGN,YSIGN,ZERO,ONE,JA,JB,JC,NRR,ITAPF1,ITAPF2
COMMON /T121/ISID,NAA,NBH,MINS,AACT(20),AICOL(20),ASIGN(20),ISIF,
* ACOFF(20),AEWROW(50),AEWSIGN(50),AEWCOEF(50),AEWCOL(50)

```

*** READ I,TO THE PLACEMENT OF NEW ACTIVITIES(NINS) IN
OLD MATRIX, NAMES OF NEW ACTIVITIES(AACT), AND COEF.
OF OBJECTIVE FUNCTION.

```

ISIE = 99999
READ (-,511) NINS
READ (-,503) NAA,(AACT(IA),IA=1,NAA)
READ (-,506) (AICOL(IB),ASIGN(IB),ACOFF(IB),IB=1,NAA)

```

*** READ I,TO THE COEF. OF CONSTRAINTS.

```

READ (-,511) NBH
READ (-,512) (AEWROW(IX),AEWCOL(IX),AEWSIGN(IX),AEWCOEF(IX),IX=
* 1,NBH)

```

```

*** INSERT THE NEW ACTIVITIES INTO AOLD# OBJ. FUNCTION.
DO 30 I=1,COL
  IF (NLS .GE. K1) GO TO 40
30 CONTINUE
40 CONTINUE
DO 45 I=1,COL-MINS
  KAD=CO, +NAA+1-K2
  KAE=CO, +1-K2
  ACT(KAD)=ACT(KAE)
45 CONTINUE
DO 47 I=1,NAA
  KAC=NLS+K2
  ACT(KAC)=AACT(K2)
47 CONTINUE

*** UPDATE THE NO. OF ACTIVITIES.
COL=CO, +NAA

*** READ IN TO THE AOLD# OBJECTIVE FUNCTION FROM ITAPE1.
REWIND ITAPE1
READ (ITAPE1,201) (SIGN(ID),COFF(ID),ICOL(ID),ID=1,NOOB)

*** START TO INSERT THE NEW ACTIVITIES INTO THE OLD OBJ. FUNC.
DO 50 I=1,NOOB
  IF (NLS .GE. ICOL(K1) .AND. NINS .LT. ICOL(K1+1)) GO TO 55
50 CONTINUE
GO TO 63

55 CONTINUE
DO 60 I=1,NOOB-MINS
  KAD = NOOB+NAA+1-KAB
  KAE = NOOB+1-KAB
  SIGN(KAD)=SIGN(KAE)
  COFF(KAD)=COFF(KAE)
  ICOL(KAD)=ICOL(KAE)+NAA
60 CONTINUE

63 CONTINUE
DO 65 I=1,NAA
  KAC = NINS + K2
  SIGN(KAC)=SIGN(K2)
  COFF(KAC)=ACOFF(K2)
  ICOL(KAC)=AICOL(K2)
65 CONTINUE

*** UPDATE THE NO. OF COEFF. OF OBJ. FUNC.(NOOB)
NOOB = NOOB+NAA

*** WRITE OUT THE ANEW# OBJ. FUNC. TO ITAPE2.
WRITE (ITAPE2,201) (SIGN(ID),COFF(ID),ICOL(ID),ID=1,NOOB)

```

```

*** DO THE INSERTION FOR CONSTRAINTS.
DO 80 AF=1,NAB
  READ (ITAPE1,201) (SIGN(IH),COEF(IH),ICOL(IH),IHEI,NOCO(IE))
  READ (ITAPE1,202) ITT1,ITT2,ITT3
*** UPDATE THE COL. COORD. OF NON-ZERO COEF. IF COL. GT. NINS.
DO 71 IE=1,NOCO(IE)
  IF (ICOL(IE) .GT. NINS) ICOL(IE)=ICOL(IE)+NAA
71 CONTINUE

DO 70 IE=1,NAB
  IF (AE.ROW(IE) .NE. IE) GO TO 70
DO 67 IF = 1,NOCO(IE)
  IF (AE.COL(IE) .GT. ICOL(IE) .AND. AFWCOL(IE) .LT. ICOL(IE+1)) GO
  * TO 68
67 CONTINUE
GO TO 65
64 CONTINUE
DO 69 IF=1,NOCO(IE)-1E
  IH = ICOL(IE)+1-IE
  SIGN(IH+1) = SIGN(IH)
  COEF(IH+1) = COEF(IH)
  ICOL(IH+1) = ICOL(IH)
69 CONTINUE
75 CONTINUE
SIGN(IH+1)=AE*SIGN(IE)
COEF(IH+1)=AE*COEF(IE)
ICOL(IH+1)=AE*ICOL(IE)

*** UPDATE THE NO. OF COEF. OF CONSTRAINT.
NOCO(IE) = NOCO(IE)+1
70 CONTINUE

*** WRITE OUT THE NEW# CONSTRAINTS TO ITAPE2.
WRITE (ITAPE2,201) (SIGN(IH),COEF(IH),ICOL(IH),IHEI,NOCO(IE))
WRITE (ITAPE2,202) ITT1,ITT2,ITT3
80 CONTINUE

*** REWIND THE PERMENANT FILE (ITAF2)
REWIND ITAPE2

*** FORMAT.
01 FORMAT (5(A1,F18.5,I4))
02 FORMAT (4(I1,A1,F18.5))
03 FORMAT (13,11A7/(3*,11A7))
06 FORMAT (5(10,A1,F11.2))
11 FORMAT (15)
12 FORMAT (4(213,A1,F13.2))
  RETURN
END

```